

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO
FAKULTETA ZA MATEMATIKO IN FIZIKO

Uroš Paščinski

**Avtomatična transkripcija
večglasnega petja**

DIPLOMSKO DELO
INTERDISCIPLINARNI UNIVERZITETNI ŠTUDIJ
RAČUNALNIŠTVA IN MATEMATIKE

MENTOR: doc. dr. Matija Marolt

Ljubljana, 2015

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V diplomski nalogi preučite področje avtomatske transkripcije glasbe. Preučite pristope, ki bi bili primerni za transkripcijo večglasnega petja, izberite najprimernejšega, ga analizirajte in implementirajte. Zgradite podatkovno zbirko posnetkov večglasnega petja s transkripcijami in na njej evalvirajte algoritem.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Uroš Paščinski sem avtor diplomskega dela z naslovom:

Automatična transkripcija večglasnega petja

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Matijo Maroltom,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela, in da
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 10. marca 2015

Podpis avtorja:

Rad bi se zahvalil celotnemu profesorskemu zboru in asistentom na Fakulteti za računalništvo in informatiko ter Fakulteti za matematiko in fiziko za strokovno in nemalokrat tudi zabavno podajanje učne snovi. Posebej bi za nastanek pričujočega dela izpostavil svojega mentorja, doc. dr. Matijo Marolta, ki se mu zahvaljujem za vse nasvete, potrpežljivost in prijaznost. Zahvala gre tudi prof. dr. Anssiju Klapuriju z univerze Queen Mary v Londonu za prispevano izvirno kodo. Zahvaljujem se kolektivu Laboratorija za grafiko in multimedije na Fakulteti za računalništvo in informatiko, da so mi omogočali dostop do strojne in programske opreme. Nikakor pa ne smem pozabiti svojih staršev, ki so mi vedno stali ob strani in mi omogočili študij. Hvala vsem!

Svojim staršem.

Kazalo

Seznam uporabljenih kratic

Povzetek

Abstract

1	Uvod	1
1.1	Opis problema	2
1.2	Razdelitev na poglavja	4
2	Ozadje problema	5
2.1	Akustika	5
2.2	Digitalno procesiranje signalov	14
3	Klapurijev algoritem	27
3.1	Splošni opis algoritma	27
3.2	Podvzorčenje	30
3.3	Kratkočasovna frekvenčna analiza	35
3.4	Beljenje spektra	37
3.5	Iskanje F0	40
4	Priprava podatkov	51
4.1	Opis podatkovne množice	51
4.2	Samodejna poravnava	52
4.3	Ročna transkripcija	53

KAZALO

5 Interaktivna vizualizacija	61
5.1 Opis aplikacije	62
5.2 Analiza napak	63
6 Rezultati	69
6.1 Preurejanje podatkov pred vrednotenjem	69
6.2 Mere vrednotenja	71
6.3 Evalvacija	72
6.4 Poskusi izboljšave	75
7 Zaključek	83
Literatura	85
A Butterworthov filter	89
A.1 Magnitudni odziv Butterworthovega filtra	90
A.2 Prevajalna funkcija, ničle in poli	90
A.3 Razvoj Butterworthovega magnitudnega odziva	92
A.4 Primer načrtovanja digitalnega filtra	94
A.5 Faza Butterworthovega filtra	99

Seznam uporabljenih kratic

kratica	angleško	slovensko
DFT	discrete Fourier transform	diskretna Fourierjeva transformacija
ERB	equivalent rectangular bandwidth	enakovredna pravokotniška pasovna širina
ERBS	ERB scale	ERB lestvica
F0	fundamental frequency	osnovna frekvenca
FFT	fast Fourier transform	hitra Fourierjeva transformacija
FN	false negative	nezaznan ton
FP	false positive	napačno zaznan ton
Hz	Hertz	Hertz ali število ciklov na sekundo
KEO	finite impulse response (filter)	končni enotin odziv
MIDI	music instrument digital interface	datotečni format za simbolni zapis glasbe
MP3	MPEG 1/MPEG 2 layer III	format digitalnega zvočnega zapisa z izgubnim stiskanjem
NEO	infinite impulse response (filter)	neskončni enotin odziv
RMS	root mean square	koren kvadratnega povprečja
STFT	short-time Fourier transform	kratkočasovna Fourierjeva transformacija
TP	true positive	pravilno zaznan ton

kratica	angleško	slovensko
ZRC SAZU	Research Centre of the Slovenian Academy of Science and Arts	Znanstvenoraziskovalni center Slovenske akademije znanosti in umetnosti ¹

¹<http://www.zrc-sazu.si>

Povzetek

V tem delu se posvetimo avtomatični transkripciji večglasnega petja. Od celotne transkripcije se omejimo na detekcijo tonov. Pripravimo si testno množico večglasnih slovenskih ljudskih pesmi, ki jih pridobimo iz terenskih posnetkov, in zanje zgradimo ročne transkripcije. Nad testno množico poženemo splošni algoritem za polifonično detekcijo tonov. Razvijemo interaktivno vizualizacijo, ki nam ponazori delovanje algoritma in olajša iskanje napak. Rezultatov algoritma ne moremo primerjati, ker je testna množica nova. Algoritem poskusimo izboljšati. Utežitveno funkcijo magnitudnega spektra nadomestimo z linearno, kar prinese slabše rezultate od prvotnih. Poskusimo tudi z dvakratnim beljenjem magnitudnega spektra, ki se obnese nekoliko bolje, a še vedno ne zadovoljivo. Z mehkejšim pristopom k vrednotenju opazimo, da je lahko vzrok težav slabe detekcije tudi problematična testna množica, saj ima precej intonančnih težav.

Ključne besede: avtomatična transkripcija, večglasno petje, slovenske ljudske pesmi, interaktivna vizualizacija, terenski posnetki.

Abstract

In this work we focus on automatic transcription of polyphonic singing. In particular we do the multiple fundamental frequency (F0) estimation. From the terrain recordings a test set of Slovenian folk songs with polyphonic singing is extracted and manually transcribed. On the test set we try the general algorithm for multiple F0 detection. An interactive visualization of the main parts of the algorithm is made to analyse how it works and try to detect possible issues. As the data set is new we cannot compare the results. Steps are made towards improvements of the algorithm. The magnitude spectrum weighting function is replaced with a simple linear function but results in the degradation of the performance. Then we try to use double spectral whitening of the magnitude spectrum which turns out more promising, but still not satisfactory. A softer evaluation criteria shows that errors in performance might be due to the problematic test set, which has lots of intonation errors.

Keywords: automatic transcription, polyphonic singing, Slovenian folk songs, interactive visualization, terrain recordings.

Poglavje 1

Uvod

Ne mine dan, da ne bi bili oblegani z raznovrstnimi zvoki. Kriki in smeh razposajenih otrok, jutranje prelestno petje ptic, besno renčanje sosedovega psa na nič hudega hotečega pismonošo, žvenketanje ključev prihajajočega sostanovalca, škripanje tečajev omare pod težo časa, živčno hupanje voznika rdeči luči na semaforju ali pa hrustljanje sveže pečene skorjice. Mnogo je različnih zvokov, za katere sploh ne vemo, kdaj smo se jih naučili. Mnoge med njimi znamo poimenovati, nekatere celo posnemati.

Vsi našteti zvoki so kratki in enostavni. V vsakdanjem življenju jih srečujemo venomer, toda za večino njih do večera že pozabimo, da so se tisti dan sploh zgodili. So del nas in so globoko ponotranjeni. Lahko pritegnejo našo pozornost, nas opozarjajo, vodijo.

Poleg ljudi se oglašajo tudi živali in se odzivajo na zvoke. Človeka od ostalih živalskih vrst odlikuje zlasti sposobnost umetniškega izražanja. Del tega izražanja je zmožnost oblikovanja kompleksnih, dolgih zvokov in to na način, ki poslušalca lahko gane, se ga dotakne.

Vprašamo se, kaj je tisto v glasbi, kar človeka gane. Morda je to melodija, ki gre hitro v uho. Lahko je to harmonija, ubrano sozvočje akordov ali pa zvočna barva glasbila. Morda je to ritem, ki izzove ples. Glasbene ustvarjalce je vedno vodil občutek za lepoto. Vse glasbene prvine, ki jih poznamo, so morale biti enkrat razvite. Ni znano, koliko je tistih, ki izzovejo lepoto, pa je

še neodkritih. Če bi znali odgovoriti na ta vprašanja, bi sebe razumeli dosti bolje, kot se. Imamo prirojen občutek za estetiko, ampak ga moramo odkriti, pogosto s poskušanjem. Veliko glasbene teorije, kot jo poznamo danes, je šlo skozi ta proces poskušanja. Nekatere stvari zvenijo naravno, druge ne.

1.1 Opis problema

Sedaj se končno posvetimo našemu cilju. Zanima nas, kako iz zvočnega posnetka pridobimo višine tonov, ki v njem nastopajo. Naj opozorimo, da je v splošnem to težek problem, tudi za ljudi. Morda lahko vsak zažvižga melodijo, včasih sliši tudi bas, notranji glasovi pa so običajno pretrd oreh. Transkripcija glasbe, to je zapis zvočnega glasbenega dogajanja s simboli, pogosto notami, je proces, ki se ga navadno lotevajo ljudje z dolgoletnim glasbenim izobraževanjem.

V procesu glasbenega izobraževanja k boljši zmožnosti transkripcije pripomore poznavanje glasbene teorije: od lestvic, intervalov in akordov kot tudi poslušanje in igranje le-teh. Bistveno je prepoznavanje in razumevanje tega, kar je slišano. To sposobnost se seveda gradi postopno s poslušanjem posamičnih tonov, zatem intervalov (skupek dveh tonov), nato akordov (skupek treh ali večih tonov). Osvojiti je potrebno tudi ritmične vzorce. Omenjene prvine bi lahko na nek način označili kot osnovne gradnike in kot take nizkonivojske.

Na višjem nivoju se opazuje zveze akordov, določa se stopnje glede na potek harmonije, išče se teme, motive, melodije, izpeljave, ponavljanja. Z iskanjem ponavljajočih se tem se določa obliko skladbe.

Hainsworth je naredil raziskavo, ki je opisana v delu [9]. Vanjo je vključil 19 anketirancev, ki so bili izučeni glasbeniki, cilj pa je bil ugotoviti, kako se lotijo transkripcij resnih glasbenih del. Kljub temu da so transkribirali dela različnih zvrsti in z različnimi cilji, je bilo pri načinu izvedbe moč opaziti podobne vzorce. Večina najprej prepozna in skicira strukturo skladbe, včasih tudi s približno označenimi ključnimi frazami. Ko je struktura določena, se

označijo akordi in basovska linija, temu pa sledi še melodija. Nazadnje se razreši še notranje glasove. Pri tem si pomagajo z igranjem inštrumenta, ponavljajočim poslušanjem ali z ugibanjem na podlagi glasbenega konteksta.

Nadalje je ugotovil, da je proces transkripcije prej zaporeden kot vzporeden, razen pri zelo preprostih skladbah. V tem pogledu se razlikuje od večine sistemov za avtomatično transkripcijo. Druga opazka je, da se ljudje lahko osredotočamo samo na določen del polifonične skladbe, ostale pa se miselno izloči. Tretja ugotovitev se navezuje na uvodne misli tega poglavja, da so nekatere stvari za ljudi samoumevne in jih zato redko obravnavamo kot del problema. Med takšne spadajo prepoznavanje glasbil, sledenje ritmu in tempu ter prepoznavanje zvrsti, ki nadalje vpliva na določena pričakovanja glede vsebine.

Čeprav pojma transkripcije glasbe nismo posebej opredelili, pa se je iz konteksta do sedaj dalo videti, da transkripcija zajema široko področje glasbene umetnosti. Za celostno transkripcijo je potrebno identificirati glasbila, prepoznati zvrst in slog, slediti ritmičnim spremembam, ugotoviti pojavitve in konce tonov, poiskati melodijo, akorde, bas, notranje glasove, slediti poudarkom, prepoznavati razlike v dinamiki (glasnosti), izluščiti artikulacijo, okraske in še in še, hkrati pa mora biti robustna, neobčutljiva na spremembe zvočnih barv, neintonančnost, napake v igranju, odmev, morebitne šume, ipd.

Zaradi obsežnosti problema transkripcije se na področju pridobivanja informacij iz glasbe obravnava manjše podprobleme, ki jih je lažje razumeti, lažje definirati, lažje implementirati in lažje ovrednotiti. Seveda pa tudi če obstajajo robustni in uspešni postopki za reševanje posamičnih podproblemov, je še vedno izziv sestaviti celosten sistem.

V tem delu se posvetimo detekciji posameznih tonov iz večglasnih skladb. Našo testno množico sestavljajo slovenske péte ljudske pesmi. Vokalna glasba je za transkripcijo poseben izziv, saj je poleg vseh običajnih prvin prisotno še besedilo. Čeprav nas besedilo sámo ne zanima, pa posamezni fonemi spreminjajo zvočno barvo, zato je glasu težje slediti kot pa kakšnemu glasbilu.

Poleg tega se zvočne barve med pevci in pevkami bistveno bolj razlikujejo kot pri istovrstnih glasbilih.

1.2 Razdelitev na poglavja

Delo je razdeljeno na več poglavij. V drugem poglavju predstavimo ozadje problema. Sprehodimo se od nastanka zvoka pa vse do frekvenčnega analizatorja v človeškem slušnem sistemu. Podobno potem skušamo doseči z metodami digitalnega procesiranja signalov, kjer zvok kot signal pretvorimo iz časovne v frekvenčno domeno. V tretjem poglavju je opisan Klapurijev algoritem, ki analizira časovni zvočni signal in iz njega po kratkih časovnih odsekih izlušči višine tonov. V četrtem poglavju je opisana priprava podatkov za testiranje. Peto poglavje opisuje interaktivno vizualizacijo delovanja glavnega dela Klapurijevega algoritma. V šestem poglavju so zbrani rezultati testiranja Klapurijevega algoritma nad testno množico podatkov in poskusi izboljšav. Sedmo poglavje je namenjeno sklepnim mislim. V dodatku je študija Butterworthovega filtra.

Poglavje 2

Ozadje problema

V tem poglavju nanizamo nekaj ugotovitev iz področja fizikalne akustike, psihoakustike in digitalnega procesiranja signalov, ki so osnovno orodje za pristop k problemu.

2.1 Akustika

Vse od nastanka nihanja izpod violinskega loka pa do interpretacije valovanja kot občuteno in prefinjeno izvedena mojstrovina je področje, ki ga pokriva akustika. Pri tem se z valovanji ukvarja fizikalna akustika, s percepcijo pa psihoakustika. V tem razdelku opišemo zvok kot nihanje, naštejemo vrste zvoka in na kratko predstavimo človeški slušni sistem.

2.1.1 Nihanje in valovanje

Kadar opazujemo neko telo iz vidika referenčnega telesa in opazovano telo relativno na referenčnega spreminja položaj, pravimo, da se opazovano telo giblje. Če je opazovano telo v primerjavi z referenčnim dovolj majhno oz. se njegovi sestavni deli premikajo homogeno, je dovolj opazovati eno samo točko telesa.

Če v nekem času opazovanja lahko zaznamo ponavljajoče se gibanje, govorimo o valovanju. Pri pravem periodičnem gibanju se opazovano telo ne-

prestano vrača v isti položaj v enakih časovnih intervalih, ne glede na izbran položaj opazovanja. Dolžino časovnega intervala, v katerem se gibanje ponovi, označimo s **periodo** τ .

Sinusno nihanje

Najpreprostejše periodično nihanje si lahko predstavljamo z utežjo, pritrjeno na vzmet. Če bi na utež v eno izmed smeri, ki je pravokotna na smer gibanja vzmeti, pritrdili svinčnik in bi na enakomerno drsečem papirju opazovali sled, ki jo riše, bi opazili sinusno krivuljo, če le bi papir drseli v smeri, pravokotni na smer nihanja. Takšno gibanje nihala je preprosto harmonično nihanje. Zaradi krivulje, ki jo opiše, pa se imenuje tudi sinusno nihanje.

Graf, ki v vsakem trenutku opiše položaj nihanja, je **oscilogram**, prikazana krivulja pa valovna krivulja. Preprosto harmonično nihanje predstavlja valovanje okrog ravnovesne točke nič. Največji odmik iz ravnovesne lege, bodisi navzgor bodisi navzdol, imenujemo **amplituda**.

Sinusno nihanje lahko opišemo tudi z opazovanjem točke na krožnici, ki jo projiciramo na premer krožnice in v vsakem trenutku beležimo položaj projicirane točke. Začetni odmik iz ravnovesne lege je povezan s **faznim kotom**, tj. začetnim kotom točke na krožnici. S spreminjanjem hitrosti kroženja vplivamo na **frekvenco** opisanega nihanja in njeno recipročno vrednost — periodo. Enota za frekvenco je Hertz ($\text{Hz} = \text{s}^{-1}$). Ena perioda je čas vrtenja krožnice za poln kot, merjena pa je v časovnih enotah.

Čeprav je preprosto sinusno nihanje vsepovsod okrog nas, pa še zdaleč ni edino, je pa osnovni gradnik vseh periodičnih nihanj. Vsako periodično nihanje namreč lahko izrazimo z vsoto preprostih sinusnih nihanj [16].

Zvok kot valovanje

Osnovni gradnik glasbe je zvok. To je mehansko valovanje nekega elastičnega sredstva, najpogosteje zraka. Izvor valovanja, zvočilo, delce zraka periodično stiska in ustvarja spremenljiv tlak, kar v zraku povzroča zgoščenine in razredčine, to nihanje delcev pa se širi v smeri razširjanja valovanja (tj. vzdolžno

ali longitudinalno valovanje), običajno vstran od zvočila.

Frekvenca valovanja je povezana z zaznavanjem višine tona, amplituda pa z jakostjo.

Sestavljeno valovanje

Omenili smo že, da lahko vsako periodično nihanje izrazimo kot vsoto enostavnih sinusnih nihanj. Opazujmo, kaj se zgodi, kadar prekrijemo dve preprosti sinusni valovanji, ki nihata z isto frekvenco. Če seštejemo dve valovanji z enako frekvenco in enako fazo, se sestavljeno nihanje ojači. Amplitudi obeh valovanj se seštejeta, medtem ko sta frekvenca in faza nespremenjeni. Torej je nihanje še vedno preprosto. Če imata valovanji nasprotni fazi (razlika je π radianov), se združeno valovanje oslabi in se v skrajnem primeru, ko imata nihanji enaki amplitudi, izniči. Če seštejemo sinusni nihanji z različnima fazama, je frekvenca sestavljenega valovanja enaka frekvencama osnovnih valovanj in je zato še vedno preprosto sinusno nihanje, vendar pa (najvišja) amplituda v splošnem zaradi razlike v fazi ni enaka vsoti najvišjih amplitud osnovnih nihanj. Nihanje, sestavljeno iz dveh sinusnih nihanj z različnima frekvencama, privede do zanimivega učinka, ki je povezan s človeškim slušnim sistemom in ga zato obravnavano nekoliko kasneje.

2.1.2 Kategorizacija zvoka

Zvok lahko glede na sestavljenost razvrstimo v štiri skupine: ton, zven, šum in pok. Kadar je zvok sestavljen samo iz preprostega sinusnega nihanja, tak zvok opišemo kot čisti **ton**. Ustvariti ga je mogoče le z elektronskim glasbilom. Če je zvok sestavljen iz več sinusnih nihanj, tj. več čistih delnih tonov, govorimo o obarvanem tonu oz. **zvenu** in je zvok, ki ga ustvarjajo glasbila ter človeški glas. Zvoka preostalih dveh skupin naštejemo kasneje.

Razmerje delnih tonov v zvenu je lahko harmonično ali neharmonično. Delni toni so v **harmoničnem** razmerju, če lahko vse frekvence delnih tonov izrazimo kot celoštevilski večkratnik najnižje, osnovne frekvence. Takšno

razmerje delnih tonov je značilno za nihanje strun in piščali. Pri **neharmoničnem** razmerju so razmerja frekvenc delnih tonov necela števila. Tipični predstavniki so zvonovi, plošče in palice.

Zven in ton sta periodični nihanji. Frekvenca, ki je pri zvenu najnižja v vrsti delnih tonov, določa višino tona in se imenuje osnovna frekvenca. Višji delni toni ton obarvajo. Ljudje zven prepoznamo kot da bi bil en sam ton — obarvan ton. Zveni različnih glasbil se med sabo razlikujejo po količini zastopanosti višjih delnih tonov. K barvi prispevajo tudi t.i. formanti, ki nastopijo zaradi lastnih frekvenc resonatorjev.

Preostali dve kategoriji zvoka sta za razliko od tona in zvena proizvedeni z neperiodičnimi zvočili. Pri **šumu** je razmerje delnih tonov neharmonično in mu je težko določiti višino. Glede na spekter zastopanosti frekvenc šume ločimo med seboj po barvi. Poslednja vrsta zvoka je kratkotrajen zvok in se imenuje **pok**. Barvo določa trajanje poka [11].

Temperirana uglasitev

Leta 1939 so na 2. mednarodni konferenci za komorni ton v Londonu določili **komorni ton** a_1 . To je postal ton s frekvenco 440 Hz.

V zahodni glasbi se uporablja dvanaajsttonski sistem. To pomeni, da je v eni oktavi dvanaest različnih tonov. Ena oktava se v frekvencah razteza od f do $2 \cdot f$ Hz. Frekvenčna razmerja med toni znotraj oktave pa so definirani z uglasitvijo.

Glasbila s tipkami so praviloma uglašena v temperirani uglasitvi, ki oktavo razdeli na 12 tonov tako, da so razmerja med sosednjimi toni enaka. Če izhajamo iz komornega tona, lahko frekvence vseh ostalih tonov podamo relativno:

$$a_1 + n = 440 \cdot 2^{\frac{n}{12}},$$

kjer n pomeni število tonov vstran od komornega tona.

Piščali, pihala, trobila in ostala glasbila, ki se uglašujejo s podaljševanjem

cevi, niso uglašena v temperirani uglasitvi, saj toni nastajajo s spreminjanjem valovne dolžine zračnega stebra, kar se naravno zgodi v harmoničnem razmerju. Pri temperirani uglasitvi takšnemu razmerju sledijo samo oktave.

2.1.3 Človeški slušni sistem

Vso naštetu zračno valovanje ne bi imelo smisla pri ljudeh, če ne bi bili sposobni zračnih tresljajev razumeti kot glasbo. Poleg procesov, ki se ob poslušanju izvajajo v možganih in ki omogočajo razločevanje šuma od Beethovna, je predpogoj pretvorba zračnih tresljajev v mehansko valovanje in šele nato v električne impulze, ki jih razumejo možgani. V tem razdelku opišemo, kako slušni sistem zvočne tresljaje spremeni v električne signale.

Zunanje in srednje uho

Zračni tresljaji v človeški slušni sistem vstopijo skozi uhelj, ki je vidni zunanji del ušesa in se nato po cevastem sluhovodu, dva do trikrat ojačani, prebijejo do bobniča v srednjem delu ušesa. Ta prenese tresljaje na tri koščice: stremence, kladivce in nakovalce, ki se nahajajo v bobnični votlini, napolnjeni z zrakom. Bobnič je zelo občutljiv in zazna že nihanja z amplitudo 10^{-9} cm. Koščice nihanje okrepijo in ga prenesejo do ovalnega okenca.

Notranje uho

Z vstopom nihanja skozi ovalno okence se začne notranji del ušesa, ki je za študijo delovanja slušnega sistema najpomembnejši. V tem delu ušesa se nahaja osnovna membrana, potopljena v tekočino, in se razteza vzdolž celotnega kanala, od ovalnega okenca do vrha polža. Membrana se odzove na nihanje, čutne celice na membrani zaznajo izbočene dele, njih vzburjenje pa privede do nastanka električnega impulza. Notranje uho deluje kot frekvenčni analizator. Pri ovalnem okencu so zaznane visoke frekvence, proti polžu pa vedno nižje.

Slušno območje

Človeški sluh zaznava frekvence med 16 Hz in 20 kHz. Najbolj je občutljiv na frekvence med 1 kHz in 3 kHz [11].

Utripanje

Če valovanje sestavimo iz dveh osnovnih sinusnih nihanj, ki imata le nekoliko različni frekvenci, f_1 in f_2 , pride do pojava, ki mu pravimo utripanje. Namesto da bi slušni sistem zaznal dva tona, zazna enega s povprečno frekvenco $\frac{f_1+f_2}{2}$. Razlog za to je omejena frekvenčna ločljivost osnovne membrane. Zaradi različnih frekvenc se tekom časa spreminja tudi razlika med fazama, kar v neki časovni točki privede do izničenja valovanja, nato pa se amplituda spet okrepi. To zaznamo kot utripanje. Če sta frekvenci f_1 in f_2 dovolj razmaknjeni, zaznamo dva različna tona [16].

2.1.4 Barkova in ERBS lestvica

Pri sistemih, ki posnemajo človeški slušni model, pogosto srečamo Barkovo ali pa ERBS lestvico, ki predpisujeta razdalje med zaporednimi frekvencami.

Kritična pasovna širina

Kritična pasovna širina (*critical bandwidth*) je za dano osrednjo frekvenco f (*central frequency*) definirana kot največja frekvenčna pasovna širina okrog f , ki v notranjem ušesu aktivira isti del osnovne membrane, in je mera sposobnosti ušesa pri razločevanju sočasnih tonov oz. delnih tonov [16].

Poskusa, s katerima so raziskovalci kritično pasovno širino izmerili, sta dveh vrst. Pri prvem poskusu so opazovali najmanjšo potrebno frekvenčno razliko med dvema delnima tonoma, da ju poslušalec lahko razloči. Problem tega pristopa je pojav utripanja, ko sta frekvenci tonov blizu skupaj, kar pri poslušalcu otežuje zmožnost razločevanja. Drug poskus tega problema nima: čisti ton so prikriili z belim šumom in opazovali, kolikšen je največji možni interval suma okrog čistega tona, da poslušalec ton še razloči (vendar

se meritev izvaja s krajšanjem tega intervala). Z rezultati drugega poskusa so opisali novo mero, povezano s kritičnim razponom, in se s kratico imenuje ERB [16, 17].

ERB

V psihoakustiki se z ERB, kratico za *Equivalent Rectangular Bandwidth*, označuje mero, ki osrednji frekvenci predpiše frekvenčno pasovno širino, in je dobljena izkustveno s posnemanjem delovanja sklopa filtrov (*filterbank*) v človeškem notranjem ušesu. Sklop filtrov je modeliran z enakovrednimi pravokotniki.

Glasberg in Moore [6] sta leta 1990 enoto ERB predstavila s približkom prvega reda, ki velja za zvoke srednje jakosti in v frekvenčnem območju od 100 Hz do 10 kHz. Aproksimacija je dana z naslednjo linearno enačbo:

$$\text{ERB}(f) = 24,7 \cdot (0,00437 \cdot f + 1), \quad (2.1)$$

kjer sta frekvenca f in $\text{ERB}(f)$ izražena v Hz. Osnovana je na predpostavki, da vsaka kritična pasovna širina ustreza konstantni razdalji vzdolž osnovne membrane.

Barkova lestvica

Frekvenco f lahko izrazimo v enotah kritične pasovne širine. Na ta način dobimo Barkovo lestvico in nam pove, koliko enot kritične pasovne širine lahko zaporedoma, neprekrivajoče in tesno skupaj zložimo med frekvencama 0 in f Hz.

ERBS lestvica

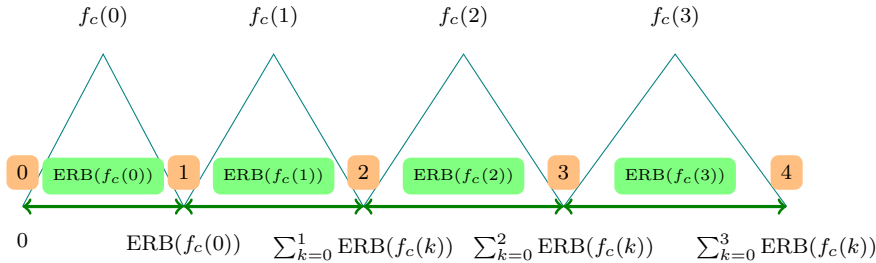
Podobno kot Barkovo lahko dobimo ERBS lestvico, če kritično pasovno širino nadomestimo z enotami ERB. Za frekvenco $f = \sum_{k=0}^n \text{ERB}(f_c(k)) + \delta f$ in $n \in \mathbb{N} \cup \{0\}$ lahko ERBS definiramo takole:

Definicija 2.1.1.

$$\begin{aligned}
\text{ERBS}(0) &= 0, \\
\text{ERBS}(f) &= \text{ERBS} \left(\sum_{k=0}^n \text{ERB}(f_c(k)) + \delta f \right) \\
&= n + 1 + \frac{\delta f}{\text{ERB}(f_c(n+1))},
\end{aligned}$$

kjer so $f_c(\cdot)$ takšne osrednje frekvence, da se pasovne širine ERB zložijo zaporedoma skupaj od frekvence 0 naprej in $0 \leq \delta f < \text{ERB}(f_c(n+1))$.

Za argumente oblike $f = \sum_{k=0}^n \text{ERB}(f_c(k))$ ERBS po zgornji definiciji 2.1.1 vedno vrne naravno število. Uvedba deleža pokritja naslednje pasovne širine v zgornji definiciji, tj. $\frac{\delta f}{\text{ERB}(f_c(n+1))}$, omogoča poizvedovanje po ERBS s poljubno frekvenco. Za lažjo ponazoritev si to lestvico lahko predstavljamo kot na sliki 2.1.



Slika 2.1: ERBS lestvica. Števila v oranžnih okvirčkih označujejo frekvenco v enotah ERB. Oznake v zelenih okvirčkih so pasovne širine okrog osrednjih frekvenc $f_c(\cdot)$.

Natančen izračun ERBS

Iz definicije ERBS je razvidno, da lahko ERBS izračunamo s poznavanjem ERB in osrednjih frekvenc $f_c(k)_{k=0}^{n+1}$. Slednje je mogoče izračunati iz nasle-

dnjega sistema enačb:

$$f_c(n) = \sum_{k=0}^{n-1} \text{ERB}(f_c(k)) + \frac{1}{2} \text{ERB}(f_c(n)),$$

od koder lahko $f_c(n)$ za vsak $n \in \mathbb{N}$ izrazimo kot rekurenčno relacijo, odvisno le od $f_c(n-1)$. Čeprav to ni potrebno, je takšno rekurzivno enačbo mogoče analitično izraziti v eksplisitni obliki, tako da je $f_c(n)$ takšna funkcija argumenta n , izračun katere ne zahteva poznavanja prejšnjih vrednosti, tj. $f_c(k)$ pri $k < n$.

Aproksimacija ERBS

Naj bo $\delta f > 0$. Opazujemo naslednjo zvezo:

$$\begin{aligned} & \text{ERBS} \left(\sum_{k=0}^n f_c(k) + \delta f \right) - \text{ERBS} \left(\sum_{k=0}^n f_c(k) \right) \\ &= n + 1 + \frac{\delta f}{\text{ERB}(f_c(n+1))} - (n + 1) = \frac{\delta f}{\text{ERB}(f_c(n+1))}. \end{aligned}$$

Po deljenju z δf se izraz prevede na

$$\frac{\text{ERBS}(\sum_{k=0}^n f_c(k) + \delta f) - \text{ERBS}(\sum_{k=0}^n f_c(k))}{\delta f} = \frac{1}{\text{ERB}(f_c(n+1))}. \quad (2.2)$$

Izraz na desni strani (2.2), opazovan kot funkcija frekvence f , zaradi omejitve na osrednje frekvence opisuje stopničasto funkcijo. Nadomestimo jo lahko z zvezno, če osrednje frekvence zamenjamo z vhodno frekvenco f , $\frac{1}{\text{ERB}(f)}$.

V limiti $\delta f \rightarrow 0$ gre leva stran (2.2) proti odvodu:

$$\begin{aligned} \lim_{\delta f \rightarrow 0} \frac{\text{ERBS}(\sum_{k=0}^n f_c(k) + \delta f) - \text{ERBS}(\sum_{k=0}^n f_c(k))}{\delta f} \\ = \frac{d \text{ERBS}(f)}{df} = \frac{1}{\text{ERB}(f)}, \end{aligned}$$

ki pa s skupaj z začetnim pogojem $\text{ERBS}(0) = 0$ predstavlja diferencialno

enačbo za izračun ERBS. Če za aproksimacijo ERB uporabimo (2.1), se rešitev glasi:

$$\text{ERBS}(f) = \int \frac{df}{24,7(0,00437f+1)} \approx 21,4 \cdot \log_{10}(0,00437 \cdot f + 1) \quad (2.3)$$

in se ujema z rezultatom v [6].

Inverz ERBS

Z obratom (2.3) vrednost v enoti ERB pretvorimo nazaj v Hz.

$$f = \frac{1000}{4,37} \left(10^{\frac{\text{ERBS}(f)}{21,4}} - 1 \right). \quad (2.4)$$

2.2 Digitalno procesiranje signalov

V prejšnjem razdelku smo zvok obravnavali kot nihanje. Od sedaj naprej pa bomo o zvoku govorili kot o signalu. Začeli bomo z obravnavo zvoka kot analogen, oz. natančneje časovno zvezen signal, nato pa bomo opravili prehod na diskretno časovno domeno in končno prevedbo v digitalen signal. Zvezen signal bomo zapisali kot vsoto preprostih sinusnih funkcij, ki vodi do transformacije časovne v frekvenčno domeno. Signal bomo opazovali skozi kratko časovno okno in omenili s tem povezane težave, nekatere od njih bomo skušali rešiti z uporabo okenskih funkcij. Kot je v elektro inženiringu običajno, bomo imaginarno komponento kompleksnega števila označevali z j in ne z i , kot je to običajno v matematični literaturi.

2.2.1 Zvok kot signal

Pri opazovanju zvoka kot nihanje smo izpostavili tri parametre, s katerimi lahko v vsakem trenutku opišemo stanje nihajočega telesa. To so amplituda, frekvenca in faza. Izkaže se, da zadostuje že en sam parameter. Zveznočasoven signal lahko predstavimo kot funkcijo časa t , katere vrednosti opišejo amplitude v času t . Označimo jo kot $x(t)$.

2.2.2 Priprava

Zaradi nekoliko bolj matematične obravnave signala v tem delu podamo definicije posebnih funkcij.

Diracova delta

Za potrebe računanja Fourierjeve transformacije definiramo Diracovo delta funkcijo.

Definicija 2.2.1.

$$\delta(x) = \begin{cases} \infty & x = 0 \\ 0 & x \neq 0. \end{cases}$$

Zanjo velja

$$\int_{-\infty}^{\infty} \delta(x) dx = 1.$$

To ni prava funkcija, saj bi bil pri tej definiciji njen integral ničeln. Ustreza pa definiciji porazdelitvene funkcije gostote in jo lahko vidimo tudi kot limito Gaussove porazdelitvene funkcije gostote z ničelno povprečno vrednostjo v parametru standardnega odklona.

Pravokotniška funkcija Π

Funkcija, s katero signal odrežemo, kot je učinek pravokotniške okenske funkcije.

Definicija 2.2.2.

$$\Pi(x) = \begin{cases} 0 & |x| > \frac{1}{2} \\ \frac{1}{2} & |x| = \frac{1}{2} \\ 1 & |x| < \frac{1}{2}. \end{cases}$$

2.2.3 Kratkočasovna frekvenčna analiza

Iz vhodnega signala v časovni domeni želimo pridobiti frekvence, ki v njem nastopajo. Za vsako frekvenco želimo tudi vedeti, kako močno je v signalu prisotna in kdaj se časovno pojavi. Dekompozicijo signala na spektralne komponente (tj. enostavne sinusoide) lahko opravimo s hitro Fourierjevo transformacijo, s katero pridobimo informacijo o frekvencah, ki v signalu nastopajo, in njihovi moči. Če želimo pridobiti še časovno informacijo, je ena možnost ta, da signal razdelimo na enake kratke časovne intervale, ki jim rečemo okvirji, vsak okvir pa analiziramo s hitro Fourierjevo transformacijo. Težava pri tem pristopu je, med drugim, najti ustrezno dolžino časovnega okvirja — daljši okvir pomeni slabšo časovno, a boljšo frekvenčno ločljivost, krajši okvir pa ravno nasprotno. Poglejmo sedaj omenjene stvari bolj podrobno.

Fourierjeva vrsta

Predpostavimo, da je $x(t)$ zvezna funkcija v odvisnosti od časa $t > 0$. Naj bo τ perioda, izražena v sekundah in t_0 začetni čas. Fourierjev izrek pravi, da lahko vsako periodično funkcijo s periodo τ , integrabilno na nekem intervalu $[t_0 - \frac{\tau}{2}, t_0 + \frac{\tau}{2}]$, zapišemo kot neskončno vrsto sinusoid, ki nihajo s frekvenco pri celoštevilskih večkratnikih osnovne frekvence $\frac{1}{\tau}$ [1]:

$$x(t) = a_0 + \sum_{n=1}^{\infty} (a_n \cos(2\pi n \frac{t}{\tau}) + b_n \sin(2\pi n \frac{t}{\tau})), \quad t \in [t_0 - \frac{\tau}{2}, t_0 + \frac{\tau}{2}], \quad (2.5)$$

kjer se koeficienti izražajo kot

$$\begin{aligned} a_0 &= \frac{1}{\tau} \int_{t_0 - \frac{\tau}{2}}^{t_0 + \frac{\tau}{2}} x(t) dt, \\ a_n &= \frac{2}{\tau} \int_{t_0 - \frac{\tau}{2}}^{t_0 + \frac{\tau}{2}} x(t) \cos(2\pi n \frac{t}{\tau}) dt, \\ b_n &= \frac{2}{\tau} \int_{t_0 - \frac{\tau}{2}}^{t_0 + \frac{\tau}{2}} x(t) \sin(2\pi n \frac{t}{\tau}) dt. \end{aligned} \quad (2.6)$$

Funkcija x je s Fourierjevo vrsto (2.5) na intervalu $[t_0 - \frac{\tau}{2}, t_0 + \frac{\tau}{2}]$ natančno izražena, zunaj tega intervala pa le, če je periodična s periodo τ . V splošnem zvočni signal x ni periodičen s periodo τ , zato nanj gledamo kot na neko splošno funkcijo. Toda če je x integrabilna na celotnem definicijskem območju, jo lahko s Fourierjevo vrsto zapišemo po posameznih intervalih. Zato si lahko predstavljamo, da časovni parameter t_0 drsimo vzdolž signala x , ki ga s Fourierjevo vrsto izrazimo pri različnih časovnih vrednostih t_0 . Pri tem za fiksno izbrano periodo τ in drseč parameter t_0 opazujemo koeficiente a_n ter b_n , skupaj z n , saj n določa frekvenco sinusoid, ki znaša $\frac{n}{\tau}$ Hz. Koeficienti a_n in b_n so amplitude sinusoid, ki so v zvočnem signalu povezane z jakostjo, frekvence $\frac{n}{\tau}$ pa z višino tonov. Koeficient a_0 predstavlja povprečno vrednost signala, kot je razvidno iz njegove definicije. Običajno želimo, da je ta vrednost nič, saj ima neničelna vrednost neželene učinke na zvočni signal [10]. Če je signal x na voljo v celoti, tj. signala ne prejemamo in procesiramo v realnem času, potem je a_0 enostavno izračunati in odšteti iz signala, sicer je treba uporabiti filtre. Kakorkoli že, a_0 v nadaljevanju izključimo iz razprave. Iz dekompozicije signala x po (2.5) lahko torej pridobimo to, kar želimo: frekvence in amplitude, ki v signalu x nastopajo ob določenem času. Preden pa ta problem prilagodimo za diskretne časovne signale, si oglejmo dve omejitvi.

Omejitvi Fourierjeve vrste

Prva omejitev je, da moramo neskončno vrsto (2.5) zamenjati s končno, saj ni niti mogoče niti praktično, da bi si lahko shranili koeficiente a_n in b_n za števno neskončno vrednosti n . To prinese s seboj dve posledici: x ne moremo več izraziti natančno, pač pa dobimo zgolj aproksimacijo, hkrati pa se nabor frekvenc, ki jih lahko izrazimo, zmanjša, saj so le-te odvisne od n . K sreči pa tudi x ne vsebuje vseh frekvenc. Še boljše, zaradi praktičnih omejitev človeškega sluha, da je omejen na frekvence od 16 do 20000 Hz, ni potrebno, da bi zvočni signal x vseboval frekvence, višje od omenjene zgornje meje. Iz tega sledi, da je smiselno obdržati samo prvih nekaj členov vrste (2.5). Naj K označuje zadnji člen končne Fourierjeve vrste.

Druga omejitev je frekvenčna ločljivost. Frekvence, izražene kot $\frac{n}{\tau}$ Hz, tvorijo končno harmonično zaporedje. S fiksno izbrano periodo τ lahko izrazimo K različnih frekvenc z ločljivostjo $\frac{1}{\tau}$ Hz. Očitno lahko s povečanjem τ , ki povzroči tudi povečanje K , dobimo daljše zaporedje bolj gosto posejanih frekvenc, ki jih lahko predstavimo. Po drugi strani pa večji τ pomeni daljši časovni interval, na katerem aproksimiramo x , zato je čas, pri katerem se frekvence v vrsti pojavijo, težje določiti. Potrebno je najti pravo ravnovesje med časovno in frekvenčno ločljivostjo, kar naredimo z izbiro parametra τ . Majhna frekvenčna ločljivost je problematična zlasti pri nizkih frekvencah, saj nizki toni ležijo bližje skupaj kot pri visokih frekvencah, kar je posledica dejstva, da frekvence tonov naraščajo v eksponentnem zaporedju (v eni oktavi se frekvenca podvoji). Nekatere metode, npr. [4], zato za omilitev tega problema uporabijo t.i. večločljivostni pristop, kjer τ prilagajajo glede na frekvenčno območje analize. Pri nizkih frekvencah uporabijo večji τ , pri višjih frekvencah pa τ zmanjšajo. Druga možna rešitev pa je uporaba t.i. valjčne transformacije [1].

Fourierjeva transformacija

Čeprav dekompozicija x s Fourierjevo vrsto dobro ustreza naši definiciji problema, pa imamo kot vhod v algoritem še vedno samo levo stran enačbe (2.5). Zanima nas, kako dobimo koeficiente a_n in b_n , skupaj s frekvencami, pri katerih nastopajo. Odgovor je s korelacijo oz. natančneje, s Fourierjevo transformacijo, ki je, nepresenetljivo, tesno povezana s Fourierjevo vrsto. Zaradi njene izjemno široke uporabne vrednosti obstaja več oblik Fourierjeve transformacije, ki so bolj ali manj enakovredne, vendar so prilagojene specifičnim področjem uporabe. Zvezno Fourierjevo transformacijo definirajmo takole:

$$X(\Omega) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi t\Omega} dt. \quad (2.7)$$

Ker je v integracijo vključena kompleksna sinusoida $e^{-j2\pi t\Omega}$, je transformacija $X(\Omega)$ kompleksno število. Kompleksna sinusoida, ki jo po Eulerjevi formuli sestavljata realen kosinus in imaginaren sinus, ima frekvenco Ω , ki pa je realno število. Opazimo lahko podobnost med Fourierjevo transformacijo (2.7) in koeficienti Fourierjeve vrste (2.6). Če koeficienta a_n in b_n zapišemo kot kompleksno število

$$\begin{aligned} a_n - jb_n &= \frac{2}{\tau} \int_{t_0 - \frac{\tau}{2}}^{t_0 + \frac{\tau}{2}} x(t) (\cos(2\pi n \frac{t}{\tau}) - j \sin(2\pi n \frac{t}{\tau})) dt \\ &= \frac{2}{\tau} \int_{t_0 - \frac{\tau}{2}}^{t_0 + \frac{\tau}{2}} x(t) e^{-j2\pi n \frac{t}{\tau}} dt, \end{aligned}$$

s pomočjo Eulerjeve enakosti podobnost postane očitna. Fourierjeva transformacija (2.7) je splosnejša od (2.6), saj je definirana za vse frekvence (Ω je poljubno realno število). Transformacija pa se ji reče zato, ker funkcijo časovne spremenljivke prevede na funkcijo frekvenčne spremenljivke.

Če zvezno frekvenco Ω nadomestimo s harmoničnim zaporedjem frekvenc $\frac{1}{\tau}, \frac{2}{\tau}, \dots, \frac{K}{\tau}$, se zdi, da bi s transformacijo (2.7) lahko uspešno izračunali koeficiente a_n in b_n za $n = 1, \dots, K$. Toda še vedno moramo pojasniti, zakaj transformacija res poišče frekvence in amplitude v signalu x . Fourierjevo transformacijo lahko vidimo kot korelacijo x s kompleksno sinusoido oz. korelacijo x z dvema sinusoidama, medsebojno zamaknjenima za fazo $\frac{\pi}{2}$. Zaradi relacije (2.5) lahko predpostavimo, da je funkcija x sestavljena iz vsote sinusoid, ki imajo različne amplitude, frekvence in faze (slednja je implicitno prisotna v koeficientih a_n in b_n). Od tod in iz linearnost integrala sledi, da je dovolj opazovati, kako se obnaša korelacija med dvema enostavnima sinusoidama. To lahko storimo z algebraično telovadbo. Za lažjo ponazoritev sledi primer.

Primer: zvezna Fourierjeva transformacije sinusoide

Naj bo x preprosta sinusoida z amplitudo a , frekvenco α in fazo ϕ , definirana za vse $t \in \mathbb{R}$ (neskončna periodična funkcija): $x(t) = a \cos(2\pi\alpha t + \phi)$.

Izračunajmo zvezno Fourierjevo transformacijo funkcije x .

$$X(\Omega) = \int_{-\infty}^{\infty} a \cos(2\pi\alpha t + \phi) e^{-j2\pi\Omega t} dt.$$

Po Eulerjevi enakosti lahko kosinus nadomestimo z razliko konjugiranih kompleksnih eksponentov: $\cos(2\pi\alpha t + \phi) = \frac{1}{2} (e^{j2\pi\alpha t + j\phi} - e^{-j2\pi\alpha t - j\phi})$. Če upoštevamo še linearnost integrala, dobimo razliko dveh integralov.

$$X(\Omega) = \frac{a}{2} \int_{-\infty}^{\infty} e^{j2\pi\alpha t + j\phi} e^{-j2\pi\Omega t} dt - \frac{a}{2} \int_{-\infty}^{\infty} e^{-j2\pi\alpha t - j\phi} e^{-j2\pi\Omega t} dt.$$

Zaradi podobnosti lahko izračunamo samo en integral.

$$\frac{a}{2} \int_{-\infty}^{\infty} e^{j2\pi\alpha t + j\phi} e^{-j2\pi\Omega t} dt = \frac{a}{2} e^{j\phi} \int_{-\infty}^{\infty} e^{j2\pi t(\alpha - \Omega)} dt.$$

Preostali del integracije pa je najbolj zvit. Integriramo kompleksno sinusoido, ki je enotska krožnica s središčem v izhodišču kompleksne ravnine. Ko t teče po realni osi, po tej krožnici krožimo. Po enem obhodu krožnice se zgodi naslednje. Če $\alpha - \Omega \neq 0$, se kompleksna števila med sabo odštejejo. Če $\alpha - \Omega = 0$, je $e^{j2\pi t(\alpha - \Omega)} = e^0 = 1$ za vse $t \in \mathbb{R}$, torej se v tem primeru enice konstruktivno seštejejo skupaj. To ravno ustreza definiciji Diracove delta funkcije $\delta(\alpha - \Omega)$. Na podoben način rešimo še drugi integral. Rezultat je

$$X(\Omega) = \frac{a}{2} e^{j\phi} \delta(\alpha - \Omega) + \frac{a}{2} e^{-j\phi} \delta(\alpha + \Omega). \quad (2.8)$$

Od tod razberemo, da signal x niha pri frekvencah $\pm\alpha$. Število $\frac{a}{2}$ je polovica amplitude originalnega signala x . Kompleksna faktorja $e^{\pm j\phi}$ predstavljata fazi dveh sinusoidnih komponent, ki sta $\pm\phi$. \triangle

Zvezna Fourierjeva transformacija sinusoide v splošnem

Zvezna Fourierjeva transformacija neskončne periodične sinusoide s frekvenco α je neničelna le pri frekvencah $\Omega = \pm\alpha$. Drugače zapisano,

$$X(\Omega) = \begin{cases} \frac{a}{2}e^{j\phi}\delta(0), & \Omega = \alpha, \\ \frac{a}{2}e^{-j\phi}\delta(0), & \Omega = -\alpha, \\ 0, & \text{sicer.} \end{cases}$$

Če bi namesto kosinusa transformirali kompleksno sinusoido pri enakem kotu in fazi, bi dobili $ae^{j\phi}\delta(\alpha - \Omega)$, tj. brez negativne frekvence in a namesto $\frac{a}{2}$. Kadar transformiramo realno funkcijo, kot je to v primeru zvočnega signala običajno, je rezultat vedno simetričen glede na $\Omega = 0$ [10].

Primer: kratkočasovna analiza

V primeru neskončne periodične funkcije je iz zvezne Fourierjeve transformacije mogoče natančno rekonstruirati amplitudo, frekvenco in fazo, kot je pokazal rezultat (2.8). Skrajšanje periodične funkcije privede do zanimivega učinka. Pokažimo ta učinek na primeru konstantne funkcije $x(t) = 1$.

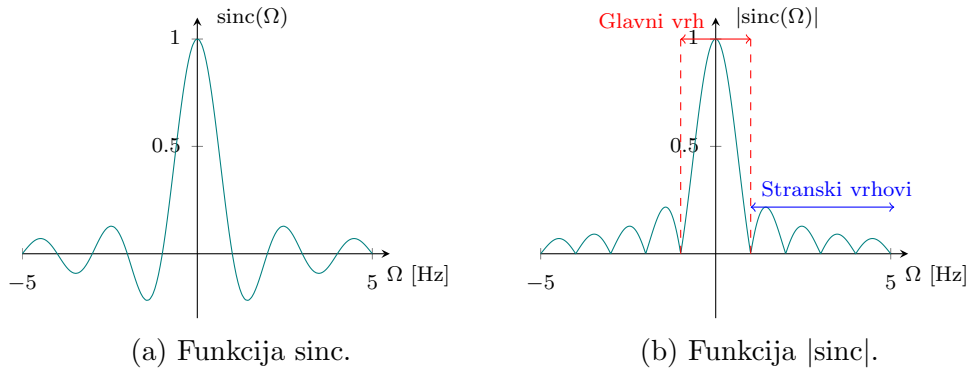
$$X(\Omega) = \int_{-\infty}^{\infty} 1 \cdot e^{-j2\pi\Omega t} dt = \delta(\Omega). \quad (2.9)$$

Konstantna funkcija ima frekvenco 0, saj je $1 = e^{-j2\pi \cdot 0 \cdot t}$. Če sedaj konstantno funkcijo omejimo na končen interval, kar lahko matematično izrazimo kot množenje konstantne funkcije s pravokotniško Π funkcijo, dobimo

$$\begin{aligned} X(\Omega) &= \int_{-\infty}^{\infty} \Pi(t)e^{-j2\pi\Omega t} dt = \int_{-\frac{1}{2}}^{\frac{1}{2}} e^{-j2\pi\Omega t} dt = -\frac{1}{j2\pi\Omega} e^{-j2\pi\Omega t} \Big|_{-\frac{1}{2}}^{+\frac{1}{2}} \\ &= \frac{je^{-j\pi\Omega}}{2\pi\Omega} - \frac{je^{j\pi\Omega}}{2\pi\Omega} = \frac{1}{\pi\Omega} \frac{j}{2} (e^{-j\pi\Omega} - e^{j\pi\Omega}) = \frac{\sin(\pi\Omega)}{\pi\Omega} = \text{sinc}(\Omega). \end{aligned}$$

Točki nezveznosti pri časih $\pm\frac{1}{2}$ povzročita razpršitev spektralne komponente okrog nosilne frekvence, kot je mogoče videti iz grafa funkcije sinc na sliki

2.2a. Podobno spektralno porazdelitev ima tudi zvezna Fourierjeva transformacija odrezane realne sinusoide, ki pa ima dva glavna vrhova (*main lobes*), simetrična glede na $\Omega = 0$. Transformacija kompleksne sinusoide je zgolj premaknjena funkcija sinc, pomnožena s konstanto. Tako torej zgledajo spektralne komponente, ko signal x analiziramo s pravokotniškim oknom.



Slika 2.2: Grafa funkcij sinc in njene absolutne vrednosti. Ko $|\Omega|$ narašča, funkcija sinc s pojemkom oscilira okrog 0. Stranski vrhovi (*side lobes*) absolutne vrednosti te funkcije v magnitudnem spektru predstavljajo problem in jih želimo zmanjšati.

Učinek stranskih vrhov želimo zmanjšati, da so glavni vrhovi v spektrogramu čimbolj izraziti. Okenske funkcije so pristop k omilitvi tega problema, čeprav zraven prinesejo kakšnega novega. \triangle

Magnitudni in fazni spektrogram

Za konec tega razdelka povejmo še to, da je v splošnem rezultat zvezne Fourierjeve transformacije kompleksno število. Ko s transformacijo analiziramo signal x za vse frekvence harmonične vrste $\frac{n}{\tau}$, $n = 1, 2, \dots, K$, dobimo K kompleksnih števil, ki jih v magnitudnem spektrogramu pri ustreznih frekvencah predstavimo z njihovimi absolutnimi vrednostmi (magnitudami), v faznem spektru pa ponazorimo argumente (faze, kote) teh kompleksnih števil.

2.2.4 Diskretnočasovna Fourierjeva transformacija

Če zveznočasovne signale izmerimo samo v določenih časovnih intervalih, dobimo diskretnočasovne signale. Zaradi enostavnosti se običajno uporablja intervale enakih dolžin. Od dolžin teh intervalov in frekvenc v signalu je odvisno, koliko vsebine se pri intervalnem zajemanju ohrani. Brž ko uvedemo diskretizacijo, se informacija o pravi zveznočasovni funkciji izgubi. Skozi opazovane točke namreč lahko napeljemo neskončno mnogo krivulj. K sreči pa nam Fourierjeva vrsta omogoča opazovanje zgolj ene družine funkcij, obenem pa lahko zgolj na osnovi podatka o času in amplitudi obnovimo frekvence in faze iz signala.

Priprava

Časovno intervalno vzorčenje signala lahko opišemo z neskončnim zaporedjem enakomerno razmaknjenih Diracovih delta funkcij.

Definicija 2.2.3.

$$\text{III}(x) = \sum_{n=-\infty}^{\infty} \delta(x - n)$$

Če časovni signal pomnožimo s III funkcijo, nam ostane diskretiziran signal. Vzorečje s periodo τ se izraža kot $\text{III}(\frac{t}{\tau})x(t)$. Izpostavimo, da je

$$\text{III}(ax) = \frac{1}{|a|} \sum_{n=-\infty}^{\infty} \delta\left(x - \frac{n}{a}\right), \quad (2.10)$$

kar sledi iz definicij III in δ .

DFT

S pomočjo funkcije III lahko zveznočasovno Fourierjevo transformacijo prevedemo na diskretnočasovno. Namesto zveznočasovnega signala $x(t)$ opazu-

jemo signal $x(t)\text{III}(t f_s)$, kjer je f_s želena frekvenca vzorčenja.

$$X(\Omega) = \int_{-\infty}^{\infty} x(t)\text{III}(t f_s)e^{-j2\pi\Omega t}dt.$$

Iz lastnosti (2.10) sledi

$$X(\Omega) = \frac{1}{|f_s|} \int_{-\infty}^{\infty} x(t)e^{-j2\pi\Omega t} \sum_{n=-\infty}^{\infty} \delta(t - \frac{n}{f_s})dt.$$

Po zamenjavi vsote in integrala dobimo

$$X(\Omega) = \frac{1}{|f_s|} \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} x(t)\delta(t - \frac{n}{f_s})e^{-j2\pi\Omega t}dt.$$

Iz definicije Diracove delta funkcije je izraz pod integralom neničeln le za $t = \frac{n}{f_s}$.

$$X(\Omega) = \frac{1}{|f_s|} \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} x(\frac{n}{f_s})|f_s|\delta(0)e^{-j2\pi\Omega \frac{n}{f_s}}dt.$$

Preuredimo v

$$X(\Omega) = \sum_{n=-\infty}^{\infty} x(\frac{n}{f_s})e^{-j2\pi\Omega \frac{n}{f_s}} \int_{-\infty}^{\infty} \delta(0)dt.$$

Po integriranju ostane

$$X(\Omega) = \sum_{n=-\infty}^{\infty} x(\frac{n}{f_s})e^{-j2\pi\Omega \frac{n}{f_s}}.$$

Iz dobljene formule opazimo, da zveznočasovni signal $x(t)$ vzorčimo s korakom f_s^{-1} . Predpostavimo, da je x končen signal. Torej obstaja $t_N > 0$, kjer je $x(t) = 0$ za $t \geq t_N$. Brez izgube za splošnost lahko privzamemo, da je $x(t) = 0$ za $t < 0$. Naj bo N število točk diskretnega signala $x[n]$, dobljenih iz zveznega $x(t)$ pri $t = \frac{n}{f_s}$ za $n \in \{0, \dots, N-1\}$. Frekvenca Ω je še vedno

zvezen parameter. Zamenjava $\Omega \leftarrow \frac{k f_s}{N}$ privede do ciljne formule za DFT.

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{nk}{N}}.$$

Digitalizacija diskretnega signala

Digitalni signal je diskreten v času, za razliko od diskretnočasovnega signala pa je diskreten tudi v amplitudah. Proces pretvorbe zveznih amplitudnih vrednosti v nek omejen diskreten nabor možnih amplitud se imenuje kvantizacija. Za domačo, neprofesionalno rabo običajno zadostuje 16-bitna kvantizacija, kar pomeni, da so amplitudne vrednosti predstavljene s 65536 različnimi števili.

2.2.5 Okenske funkcije

Stranske vrhove v magnitudnem spektru, do katerih pride zaradi rezanja signala s časovnim oknom, želimo zmanjšati. To lahko dosežemo z okenskimi funkcijami. Kot smo videli na primeru, do tega pojava pride zaradi nenadnega padca signala na nič. Ideja okenskih funkcij je narediti postopen prehod iz polnega signala v ničelni signal. Najenostavnejši primer takšne okenske funkcije je trikotniško okno z vrhom na sredini okna in simetričnem na obe polovici.

Trikotniško okno ima majhno ploščino, ravno polovico ploščine pravokotniškega okna, zato iz signala zajame pol manj energije. Če želimo na sredini okna zajeti več energije, iščemo druge funkcije, ki imajo na sredini bolj postopen padec, blizu meja okna pa bolj nenadnega, razen čisto pri mejah, ko se padanje zopet umiri.

Z uvedbo naslednje parametrizacije lahko opišemo družino pogosto uporabljenih okenskih funkcij:

$$\alpha (1 + \beta \cos(\gamma n)).$$

Za okna velikosti K vzorcev podamo nekaj primerov okenskih funkcij. Če izberemo $\alpha = 0$, $\beta = 1$ in $\gamma = 0$, opišemo pravokotniško funkcijo. Parametri $\alpha = \beta = 0,5$, $\gamma = \frac{2\pi n}{K-1}$ opišejo Hannovo okno, $\alpha = 0,54$, $\beta = 0,46$ in $\gamma = \frac{2\pi n}{K-1}$ so parametri Hammingovega okna.

Vpliv okenskih funkcij na signal lahko preučujemo z opazovanjem magnitude odziva oken. Običajno nas zanima relativna razlika med velikostjo glavnega vrha in najvišjega stranskega vrha. Negativna posledica uporabe oken je širjenje glavnega vrha. Okna, ki dobro zadušijo stranske vrhove, imajo širši glavni vrh kot tista, ki stranske vrhove dušijo slabše.

Poglavje 3

Klapurijev algoritem

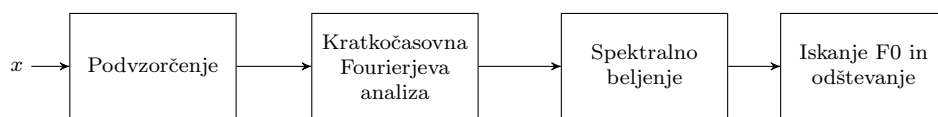
Ena izmed najpomembnejših prvin v zahodni glasbi je višina tonov. Skupaj z dolžino tonov tvorita najosnovnejše ogrodje za opis kompleksnejših struktur v glasbi. Tako je npr. melodija zgrajena iz zaporedja tonov, harmonija pa je določena z njihovo sozvočnostjo, pri obeh pa višina tonov igra osrednjo vlogo. Pridobivanje informacij iz zvočnega posnetka je zato pogosto in v veliki meri odvisno od uspešnosti določanja višine tonov, vendar je avtomatičen proces določanja višin težaven, še posebej pri polifonični glasbi. V tem poglavju se soočimo s problemom določanja višin tonov in opišemo algoritem, ki ga poskuša rešiti.

3.1 Splošni opis algoritma

Klapurijev algoritem [8], poimenovan po njegovem avtorju, je algoritem za iskanje osnovnih frekvenc (F_0) in deluje v frekvenčni domeni. Izvorno kodo je prispeval avtor sam in se v določenih delih nekoliko razlikuje od postopka, opisanega v citiranem članku, razkriva pa tudi nekatere podrobnosti. Opis v tem razdelku se nanaša na to izvorno kodo. Napisana je v programskem jeziku C++. Zaradi poenostavitve je bila v celoti prepisana v MATLAB¹.

Predpostavimo, da algoritem na vhodu prejme digitaliziran zvočni si-

¹<http://www.mathworks.com>



Slika 3.1: Shematični prikaz Klapurijevega algoritma.

gnal v časovni domeni. Iskanje osnovnih frekvenc poteka v frekvenčni domeni. Potek algoritma shematično prikazuje slika 3.1. Delovanje algoritma lahko razčlenimo na dva dela. Najprej za posamezne kratkočasovne okvirje izračuna utežen magnitudni spekter, nato v spektru poišče osnovne frekvence. Izračun magnitudnega spektra lahko strnemo v tri korake:

1. decimiranje oz. podvzorčenje vhodnega časovnega signala,
2. transformacija signala v časovno–frekvenčno domeno, katero algoritem opravi s kratkočasovno Fourierjevo transformacijo in
3. beljenje spektra v vsakem izmed okvirjev, tj. utežitev magnitudnega spektra po vzoru slušnega sistema pri ljudeh.

Čeprav je za pridobitev magnitudnega spektra nujen samo drugi korak, pa je koristno opraviti tudi podvzorčenje, kar se opazi pri izboljšani časovni zahtevnosti računanja Fourierjeve transformacije, in pa spektralno beljenje, ki poveča robustnost sistema na različne zvočne vire, posledica pa je natančnejše ocenjevanje osnovnih frekvenc.

V magnitudnem spektru je izražena moč prisotnosti posameznih frekvenc v danem kratkem časovnem okviru. Povsem intuitivno je, da bi kandidate za F0 lahko predstavljajo prvih nekaj frekvenc z največjo magnitudo, vendar bi na ta način povsem zanemarili barvno informacijo zvena. Pri glasbi z akustičnimi inštrumenti in vokali je zato smiselno upoštevati tudi delne tone. Avtor algoritma [8] je zato po zgledu v predhodnem delu nekaterih drugih raziskovalcev določanje osnovne frekvence iz spektra zastavil kot naslednji optimizacijski problem: kandidat za F0 je tista frekvenca f , kjer je dosežena

največja vrednost kriterijske funkcije

$$s(f) = \sum_{m=1}^M w(f, m) |Y(f, m)|, \quad (3.1)$$

kjer f, m za $m \in \{1, 2, \dots, M\}$ predstavlja frekvenco m -tega harmoničnega delnega tona (tj. f, m Hz), $|Y|$ označuje magnitudni spekter danega časovnega okvira vhodnega signala, w pa utežitveno funkcijo, ki posameznemu delnemu tonu predpisuje pomembnost. Koliko delnih tonov upoštevamo v izračunu (tj. velikost parametra M), kakšna je funkcija w in kako je bila dobljena, je opisano kasneje.

Na kriterijsko funkcijo (3.3) lahko gledamo kot na cenilko osnovnih frekvenc. Določanje osnovne frekvence na podlagi soobstoja harmoničnih delnih tonov v določenem razmerju glede na osnovno frekvenco se vsekakor zdi smiselno, zlasti pri tisti vrsti zvokov, pri katerih je razmerje frekvenc v zvenu harmonično. Tovrstne zvoke na primer ustvarjajo nihanja strun, pihala in trobila. Po drugi strani pa formula (3.3) ne upošteva neharmoničnih delnih tonov, ki nastopajo v zvenu teles, ki nihajo v treh razsežnostih, kamor spadajo npr. zvonovi, ploščice in palice. Z računskega vidika je smiselna, saj Fourierjev izrek pravi, da je vsako periodično funkcijo s periodo T mogoče izraziti kot superpozicijo (linearno kombinacijo) enostavnih nihanj, tj. sinusov in kosinusov, s frekvencami $\frac{k}{T}$, kjer je $k \in \{1, 2, \dots\}$. Drugače povedano, v Fourierjevi vrsti nastopajo sinusoidne funkcije v harmoničnem razmerju, tako kot so tudi v formuli (3.3) spektralne komponente v harmoničnem razmerju.

Iz (3.3) ni neposredno razvidno, kako iz magnitudnega spektra pridobiti več osnovnih frekvenc, kadar imamo opravka s polifoničnim zvočnim virom. Klapuri je v svojem članku [8] opisal tri različne metode: direktna metoda, iterativno ocenjevanje in odštevanje ter združena metoda. Recimo, da za trenutni časovni okvir vse tri metode vrnejo P osnovnih frekvenc (stopnja razpoznanje polifonije je P). Direktna metoda izračuna (3.3) za nek nabor osnovnih frekvenc $f \in [f_{\min}, f_{\max}]$ in vrne P najboljših lokalnih maksimumov. Iterativna metoda poišče tisto osnovno frekvenco f , pri kateri

je dosežena maksimalna vrednost $s(f)$, in jo skupaj z delnimi harmoničnimi toni odšteje iz spektra. Postopek ponavlja na preostanku spektra, dokler ne najde P različnih osnovnih frekvenc. Združena metoda združuje idejo direktne in iterativne metode. Najprej po (3.3) izračuna $P_1 \gg P$ lokalnih maksimumov (kot direktna metoda), nato s posebno kriterijsko funkcijo in iterativnim postopkom poišče P najboljših kandidatov za osnovne frekvence. Več o posameznih delih algoritma sledi v preostanku tega poglavja, a izmed treh omenjenih metod je opisana in implementirana samo iterativna metoda.

3.2 Podvzorčenje

Prvi korak Klapurijevega algoritma je podvzorčenje (ali decimiranje) in je na tem mestu podrobno opisano.

3.2.1 Priprava

Naj f_s označuje število vzorcev na sekundo in naj bo $N \in \mathbb{N}$ število vseh vzorcev v signalu x . Implementacija Klapurijevega algoritma deluje samo za vhodne signale s frekvenco vzorčenja, ki je enaka standardu pri glasbenih CD ploščah, in sicer $f_s = 44,1$ vzorcev na sekundo, zato predpostavimo, da je na vходу podan signal x , ki tej omejitvi ustreza. Namen podvzorčenja je zmanjšanje števila vzorcev v signalu, to pa posledično vodi do izboljšane časovne zahtevnosti pri nadaljnjem procesiranju signala. Hkrati s številom vseh vzorcev je potrebno zmanjšati tudi frekvenco vzorčenja za isti faktor, sicer bi npr. ob predvajanju decimiranega zvočnega signala x pri nespremenjeni frekvenci vzorčenja poleg hitrejšega tempa glasba zvenela višje. Podvzorčenje opišemo s štirimi koraki:

1. Izberemo čimvečji faktor podvzorčenja $d \in \mathbb{N}$, kolikor to dopušča aplikacija. Omejitev določa najvišja frekvenca, ki jo signal lahko vsebuje in se pri podvzorčenju zmanjša iz $\frac{f_s}{2}$ na $\frac{f_s}{2d}$ Hz.

2. Načrtujemo nizkopropustni filter tako, da iz signala odstrani frekvence, višje od $\frac{f_s}{2d}$ Hz.
3. Signal x sfiltriramo, rezultat je signal y z odstranjenimi visokimi frekvencami, nova frekvenca vzorčenja je $\frac{f_s}{d}$ vzorcev na sekundo.
4. Izvedemo podvzorčenje. V signalu y obdržimo vzorce $y[1 + kd]$, kjer je $k \in \{0, 1, \dots, \lfloor \frac{N-1}{d} \rfloor\}$.

Naveden postopek deluje v teoriji, v praksi je potrebno nekoliko več previdnosti. Pri potiskanju parametra d do teoretične Nyquistove meje bo največja frekvenca znašala $\frac{f_s}{2d}$ Hz in bo opisana zgolj z dvema vzorcema na periodo. To je pri šumnem signalu nezadovoljivo. Do šuma v signalu pride zaradi mnogih dejavnikov, npr. pri zajemanju zvoka z mikrofoni v nestudijskem okolju, kjer so prisotni nezaželeni zvoki iz ozadja; pri pretvorbi mehanskega zvoka v električni signal v mikrofону; zaradi interference in slabenja električnega signala, ki potuje po kablilih in skozi razne naprave; pri pretvorbi električnega signala kot nosilca zvočne informacije v digitalni signal v zvočni kartici zaradi netočnega urinega generatorja in kvantizacije; zaradi izgubnega stiskanja glasbenih datotek v formate kot so mp3, ogg in wma. Seznam se gotovo ne konča tu. V literaturi se zato predlaga, da je najvišja frekvenca, ki je za aplikacijo smiselna, določena vsaj s tremi ali štirimi vzorci [1, str. 495].

3.2.2 Načrtovanje filtra

Definirajmo frekvenčni razpon, primeren za naš problem. Z osnovnimi frekvencami v območju od 40 do 2100 Hz zajamemo tone od E_1 (E v kontra oktavi) do c_4 . Za primerjavo, pevski obseg šolanih glasov sega od najnižjega basa, tona D (v veliki oktavi), do najvišjega soprana, tona e_3 . Izraženo v frekvencah je to nekje od 70 do 1320 Hz. Netrenirani glasovi imajo tipično manjši obseg. Upoštevati je potrebno še delne tone, zaradi katerih se frekvenčni razpon raztegne v višje frekvence. Klapuri je število delnih tonov omejil na 20 (v formuli (3.3) je $M = 20$), vendar največ do frekvence okrog

6,3 kHz. To pomeni, da za osnovne frekvence v zgornjem mejnem območju algoritem upošteva samo tri delne tone, v primeru netreniranih vokalov pa vsaj šest. Teoretično bi lahko faktor decimiranja nastavili na tri vzorce ($\frac{f_s}{2d} = \frac{44100}{6} = 7350 \text{ Hz}$), v praksi je zaradi zgoraj omenjenih razlogov varneje izbrati $d = 2$, $\frac{f_s}{4} = 11025 \text{ Hz}$.

Sedaj lahko sestavimo filter. Izberimo vrsto filtra in določimo frekvenčna območja prepustnosti in neprepustnosti. Očitno želimo ohraniti nizke frekvence in zavreči visoke, zato naj bo filter nizkoprepustni. Na osnovi definirane frekvenčnega območja v prejšnjem odstavku lahko podamo specifikacije filtra. Naj $|H(e^{jf \frac{2\pi}{f_s}})|$ označuje magnitudni odziv filtra pri frekvenci f , izraženi v Hz. Potem naj filter s prevajalno funkcijo $H(z)$ zadošča naslednjim omejitvam:

$$1 \geq |H(e^{jf \frac{2\pi}{f_s}})| \geq 0,9477, \quad 0 \leq f \leq 6300 \text{ Hz}, \quad (3.2a)$$

$$0 \leq |H(e^{jf \frac{2\pi}{f_s}})| \leq 0,0371, \quad 11025 \leq f \leq 22050 \text{ Hz}. \quad (3.2b)$$

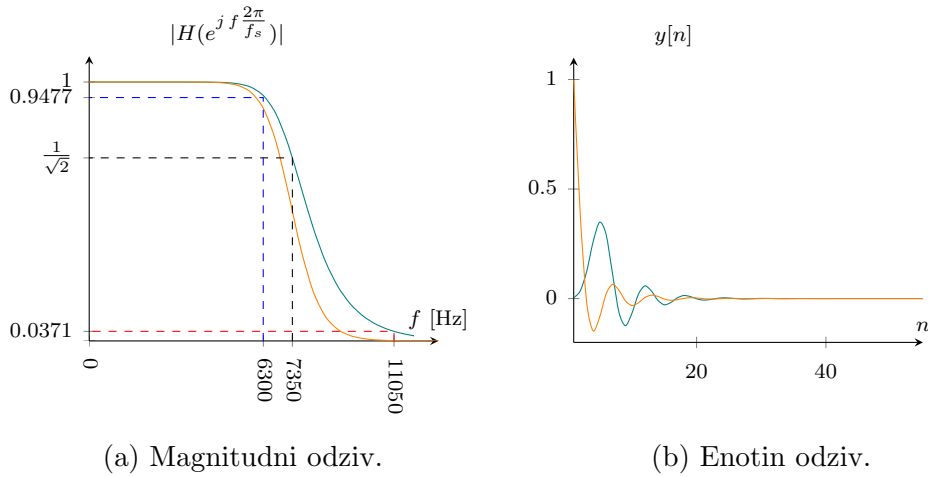
Frekvenčno območje prepustnosti se razteza od 0 do 6,3 kHz, območje neprepustnosti pa od 11025 Hz naprej.

Filtriranje lahko naredimo učinkovitejše, če posežemo po NEO filtru. Toda tovrstni filtri za razliko od KEO filtrov z (anti)simetričnimi koeficienti (tj. (anti)simetričnim enotnim odzivom) praviloma nimajo linearne faze (kaj to pomeni za filtriran signal, je razloženo v poglavju A.5). K sreči lahko fazo, ki jo v filtriran signal uvede filter, izničimo — signal je potrebno filtrirati v obe smeri. Klasične vrste NEO filtrov so Butterworthov, Čebišev tipa I, Čebišev tipa II in eliptični. Ker je v našem primeru med območjema prepustnosti in neprepustnosti dokaj široko prehodno območje, si lahko omislamo Butterworthov filter, katerega magnituda frekvenčnega odziva, kadar je uporabljen kot nizkoprepustni filter, v celotnem frekvenčnem območju monotono pada (v nobenem frekvenčnem območju nima valovanja), posledično pa je prehodno območje bolj postopno in položno kot pri Čebiševem ali eliptičnem filtru. Izračun sistemske (prevajalne) funkcije $H(z)$ z bilinearno transformacijo se nahaja v razdelku A.4, dobljena časovna diferenčna enačba pa se

glasi

$$\begin{aligned}
 y[n] = & 0,004311 x[n] + 0,025863 x[n-1] + 0,064658 x[n-2] + 0,086211 x[n-3] \\
 & + 0,064658 x[n-4] + 0,025863 x[n-5] + 0,004311 x[n-6] + 1,9816 y[n-1] \\
 & - 2,2524 y[n-2] + 1,4693 y[n-3] - 0,5963 y[n-4] + 0,1354 y[n-5] \\
 & - 0,0136 y[n-6].
 \end{aligned} \tag{3.3}$$

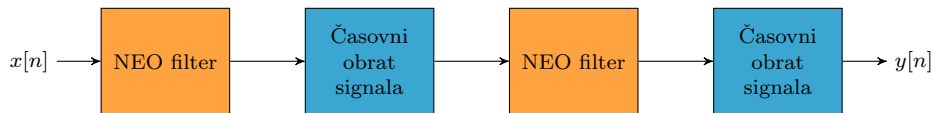
Slika 3.2 prikazuje magnitudni in enotin odziv filtra. Enotin odziv se hitro približuje ničli, kar kaže na stabilnost filtra. Iz grafov je tudi razvidno, da dvosmerno filtriranje ne vpliva le na fazo, temveč tudi na magnitudni in enotin odziv. Dvosmerno filtriranje namreč na magnitudni odziv učinkuje kot kvadriranje magnitudnega odziva, zato se magnitudni odziv zmanjša, neželena posledica pa je prekoračitev specifikacij v območju prepustnosti, vendar pa vpliv ni znaten. Pri mejnem območju prepustnosti $f = 6300$ Hz je faktor zmanjšanja magnitude približno 0,1, pri $f = 6000$ Hz pa le še 0,05.



Slika 3.2: Magnitudni in enotin odziv digitalnega Butterworthovega filtra reda 6. Modra krivulja prikazuje odziv pri običajnem enosmernem filtriranju, oranžna pa odziv pri filtriranju z ničelno fazo (dvosmerno filtriranje).

3.2.3 Dvosmerno filtriranje

Zadnja dva koraka podvzorčenja, kot že omenjeno zgoraj, sta filtriranje in zmanjšanje števila vzorcev v signalu. Postopek zmanjšanje števila vzorcev je opisan že zgoraj, 4. točka v razdelku 3.2.1, zato opišemo samo filtriranje. Filtriranje z ničelno fazo lahko opravimo po postopku [10, str. 518], prikazanem na sliki 3.3. Vhodni signal $x[n]$ sfiltriramo po enačbi (3.3), ki na izhod da $y_1[n]$. Signal $y_1[n]$ časovno obrnemo v $y_2[n]$, tako da je $\forall k : y_2[N-k] = y_1[k]$. Nato filtriranje in časovni obrat ponovimo na enak način. Rezultat je signal $y[n]$ z ničelno fazo in zmanjšanimi magnitudami neželenih frekvenc. Vsaka izmed škatlic na sliki 3.3 v posamezne spektralne komponente izhodnega signala prišteje lastno fazo, vendar časovni obrat signala fazo konjugira, zaradi česar se v celotnem postopku vpeljane faze odštejejo. V dejanski implementaciji dvosmernega filtriranja časovno obračanje ni potrebno, če rutino filtriranja prilagodimo tako, da zna filtrirati v obe smeri: od začetka proti koncu in od konca proti začetku signala.



Slika 3.3: Filtriranje z ničelno fazo. Povzeto po [10].

3.2.4 Implementacija

```

1 [y, newfs] = function decimate(x, fs)
2     [B, A] = butter(6, 1/3);
3     y = filtfilt(B, A, x);
4     y = y(1:2:end);
5     newfs = fs/2;
6 end
  
```

Izvorna koda 3.1: Podvzorčenje.

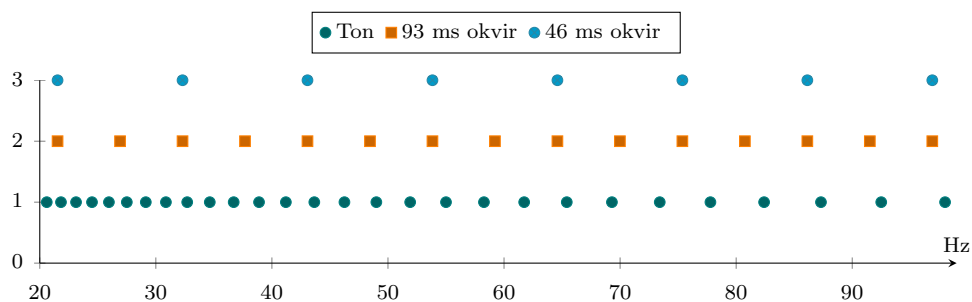
Čeprav ima postopek podvzorčenja v ozadju kar nekaj teorije, pa je implementacija v MATLABU enostavna, kot prikazuje izvorna koda 3.1. Najtežji del pri tem je seveda ugotoviti ustrezne parametre, ki pa so povezani z našeto teorijo. Funkcija `decimate` sprejme vhodni signal \mathbf{x} in frekvenco vzorčenja f_s , na izhodu pa vrne sfiltriran signal \mathbf{y} in novo frekvenco vzorčenja newfs . V drugi vrstici izračunamo koeficiente Butterworthovega digitalnega filtra reda 6 in frekvenco rezanja $\omega_c = \frac{\pi}{3}$, ki je izračunana po (A.14). Funkcija `filtfilt` opravi dvosmerno filtriranje, zadnji dve vrstici razpolovita število vzorcev in frekvenco vzorčenja.

3.3 Kratkočasovna frekvenčna analiza

Potem, ko smo iz signala odstranili polovico vzorcev, opravimo frekvenčno analizo. Tega problema se lahko lotimo na različne načine. En pristop je posnemanje človeškega slušnega sistema, še posebej delovanje notranjega ušesa, za katerega bi pripravili sklop različnih pasovno prepustnih filtrov, vsakega za nek majhen frekvenčni razpon. Skozenj bi poslali zvočni signal in v vsakem frekvenčnem območju poiskali prevladujočo frekvenco, npr. z avtokorelacijo ali pa s kratkočasovno Fourierjevo transformacijo. Informacije o frekvencah iz različnih frekvenčnih območij bi nato združili skupaj, kar bi bila mera za prisotnost višin [4]. Drug možni pristop pa je kratkočasovna Fourierjeva transformacija, ki je bila uporabljena tudi v Klapurijevem algoritmu in je opisana v poglavju 2.2.3. V nadaljevanju tega razdelka podamo parametre algoritma, ki opravi frekvenčno analizo, in omejitve, ki so posledica teh.

Zvočni signal razdelimo na kratke časovne okvirje. Klapuri je izbral dve dolžini, in sicer 46 in 93 ms, kar se pri frekvenci vzorčenja $f_s = 22050$ vzorcev na sekundo prevede na $K = 2^{10}$ oz. $K = 2^{11}$ vzorcev. Sosedna okvirja se polovično prekrivata; razdalja med sosednima okvirjema je 23 oz. 46,5 ms. Za zmanjšanje stranskih vrhov je uporabljeno Hannovo okno. Frekvenčni okvir je razširjen z ničlami, tako da je njegova velikost dvakrat večja od prvotnega okvirja. V številu vzorcev to znaša $K_0 = 2^{11}$ oz. $K_0 = 2^{12}$.

Frekvenčna ločljivost pri naštetih parametrih znaša 10,8 pri krajšem oz. 5,4 Hz pri daljšem oknu. Pri obeh dolžinah je frekvenčna ločljivost pri nizkih frekvencah preslaba, kot je to razvidno na sliki 3.4. Šele nekje nad 100 Hz postane ločljivost pri daljšem okvirju zadovoljiva, pri krajšem pa še kasneje. V območju od 20 do 100 Hz pri vokalni glasbi najdemo nizke tone baritona in basa, zato so pri vrednotenju deležni posebne pozornosti. Vendar pa je iz mogoče opaziti, da višji harmonični delni toni, ki jih je lažje natančno določiti, pomagajo pri določanju osnovne frekvence. Dodatno pa še, kot bomo videli kasneje, se lahko izračuna pri boljši ločljivosti, kot jo omogočajo frekvenčni koši magnitudnega spektra.



Slika 3.4: Frekvenčna ločljivost pri nizkih tonih v frekvenčnem območju od 20 do 100 Hz. V spodnji vrstici so prikazane frekvence tonov po formuli $440 \cdot 2^{\frac{n}{12}}$ (tj. n poltonov vstran od komornega tona a_1 s frekvenco 440 Hz). V sredini je frekvenčna ločljivost okvirja dolžine 93 ms, na vrhu pa okvirja dolžine 46 ms.

3.3.1 Implementacija

Izračun magnitudnega spektra s pomočjo kratkočasovne Fourierjeve transformacije prikazuje izvorna koda 3.2 v MATLABU. Funkcija `magSpec` sprejme zvočni signal `x`, velikost okvirja `frameSize`, z ničlami razširjeno velikost okvirja `frameSizePad`, razdaljo med sosednima okvirjema `hopSize` in največje število frekvenčnih košev `maxBin`, ki je nastavljeno tako, da najvišja frekvenca v magnitudnem spektru znaša okrog 6,3 kHz. V tretji vrstici pripravi Hanovo okno, temu pa sledi izračun kratkočasovne Fourierjeve transformacije

in nazadnje še normaliziranega magnitudnega spektra.

```
1 absX = function magSpec(x, frameSize, frameSizePad, ...  
    hopSize, maxBin)  
2     % hann(ing)  
3     hw = window(@hann, frameSize);  
4     % stft  
5     X = spectrogram(x, hw, frameSize-hopSize, frameSizePad);  
6     % magnitude spectrum (only up to maxBin needed)  
7     absX = abs(X(1:maxBin, :)) / sqrt(frameSize);  
8 end
```

Izvorna koda 3.2: Kratkočasovna Fourierjeva transformacija

3.4 Beljenje spektra

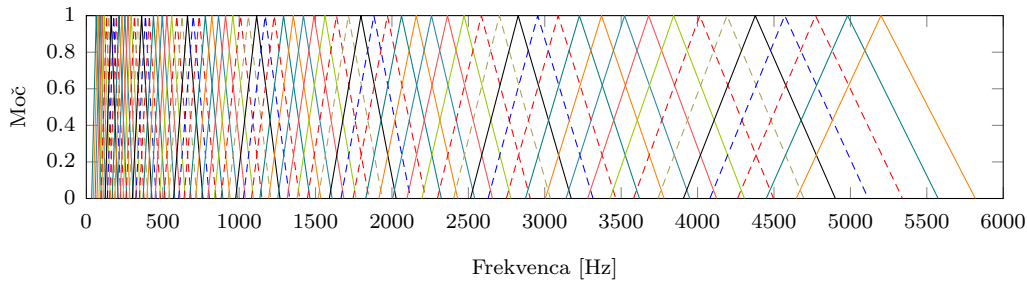
Različni zvočni viri in druge zvočne raznolikosti istega tona predstavljajo barvo zvoka in kažejo svoj odsev v frekvenčni porazdelitvi, ki tak zvok sestavlja. Zaradi barve je prepoznavanje osnovne frekvence težje, zato je njen vpliv v spektru dobro zmanjšati. Spektralno beljenje je proces ocenjevanja in odstranjevanja barvne informacije iz spektra.

Nekateri modeli spektralnega beljenja posnemajo človeški slušni sistem, zlasti delovanje notranjega ušesa. Eden takšnih je sklop delno prekrivajočih se pasovno prepustnih filtrov [12]. Pasovna širina teh filtrov je tesno povezana s kritično pasovno širino, definirano v poglavju 2.1.4. V frekvenčni domeni je tak sklop filtrov mogoče enostavno implementirati kot množenje spektra z njihovimi magnitudnimi odzivi.

Za barvo, prisotno v signalu, si lahko predstavljamo, da je bila signalu dodana skozi proces filtriranja. Če poznamo model filtra, lahko barvno informacijo izločimo z inverznim filtriranjem. Zvočno barvo lahko ocenimo iz porazdelitve spektralne energije.

V algoritmu je beljenje spektra implementirano v frekvenčni domeni, kjer se poskuša simulirati sklop pasovno prepustnih filtrov. Ti imajo v

močnostnem spektru odziv v obliki trikotnika. Sklop vsebuje 72 pasovno prepustnih filtrov s središči v osrednjih frekvencah, enakomerno razporejenih po ERB lestvici. Odzivi sklopa filtrov v močnostnem spektru so prikazani na sliki 3.5. Osrednje frekvence si lahko poljubno izberemo, razmak med



Slika 3.5: Močnostni odziv sklopa pasovno prepustnih filtrov.

njimi pa ustreza ERB lestvici. Klapuri je izbral najnižjo frekvenco 65 Hz in najvišjo 5200 Hz, vmesne pa lahko izračunamo. Razmak med sosednjima osrednjima frekvencama po ERB lestvici lahko izračunamo po (2.3) in znaša

$$\frac{1}{72-1}(\text{ERBS}(5200) - \text{ERBS}(65)) = 0,3818 \text{ ERB}.$$

Osrednja frekvenca filtra $b = 1, 2, \dots, 72$ znaša

$$\text{ERBS}(65) + (b - 1) \frac{1}{72-1}(\text{ERBS}(5200) - \text{ERBS}(65)) \text{ ERB}. \quad (3.4)$$

Z uporabo inverzne ERBS formule (2.4) lahko (3.4) pretvorimo v Hz, od tod pa po $\frac{f K_0}{f_s}$ v frekvenčni indeks diskretne Fourierjeve transformacije. Spodnja in zgornja frekvenčna meja pasovno prepustnega filtra b , ki ju zaporedoma označimo s $f_{b,\perp}$ in $f_{b,\top}$, se za vsako osrednjo frekvenco $f_{b,c}$ (v Hz) izračunata

s pomočjo inverzne ERBS enačbe (2.4):

$$f_{b,\perp} = \frac{1000}{4,37} \left(10^{\frac{\text{ERBS}(f_{b,c})-1}{21,4}} - 1 \right) \text{ Hz},$$

$$f_{b,\top} = \frac{1000}{4,37} \left(10^{\frac{\text{ERBS}(f_{b,c})+1}{21,4}} - 1 \right) \text{ Hz}.$$

Frekvence pretvorimo v frekvenčne indekse kot $i = f \cdot \frac{K_0}{f_s}$ in z ustreznim zaokroževanjem. Močnostni odziv filtra b je trikotnik $H_b(k)$, v spektru opisanem s koordinatami $\{(\lceil f_{b,\perp} \cdot \frac{K_0}{f_s} \rceil, 0), (\lfloor f_{b,c} \cdot \frac{K_0}{f_s} \rfloor + 0,5, 1), (\lfloor f_{b,\top} \cdot \frac{K_0}{f_s} \rfloor, 0)\}$. Spektralna energija se izračuna kot neke vrste standardna deviacija (utežena RMS):

$$\sigma_b = \left(\frac{1}{K_0 - 1} \sum_k H_b(k) |X(k)|^2 \right)^{\frac{1}{2}}, \quad (3.5)$$

kjer je $|X(\cdot)|^2$ močnostni odziv signala $x[n]$. Uteži $H_b(k)$ za vmesne točke

$$k \in \{\lceil f_{b,\perp} \cdot \frac{K_0}{f_s} \rceil, \dots, \lfloor f_{b,\top} \cdot \frac{K_0}{f_s} \rfloor\}$$

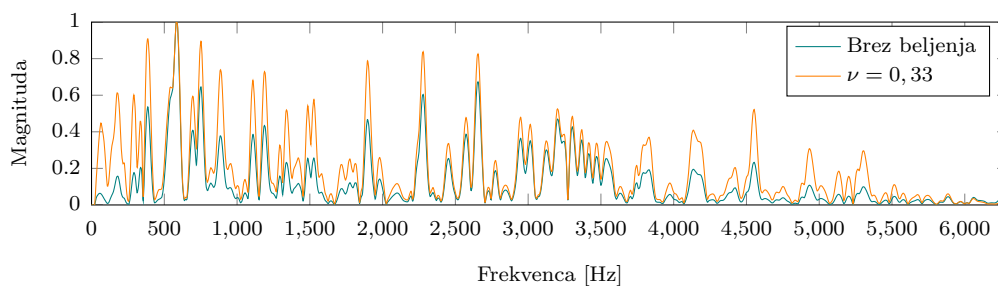
linearno interpoliramo, tj. tečejo po stranicah trikotnika filtra b . Natančneje,

$$H_b(k) = \begin{cases} \frac{k - f_{b,\perp}}{f_{b,c} - f_{b,\perp}}, & \text{če } k \in \left\{ \lceil f_{b,\perp} \cdot \frac{K_0}{f_s} \rceil, \dots, \lfloor f_{b,c} \cdot \frac{K_0}{f_s} \rfloor \right\}, \\ \frac{k - f_{b,\top}}{f_{b,c} - f_{b,\top}}, & \text{če } k \in \left\{ \lceil f_{b,c} \cdot \frac{K_0}{f_s} \rceil, \dots, \lfloor f_{b,\top} \cdot \frac{K_0}{f_s} \rfloor \right\}, \\ 0, & \text{sicer.} \end{cases} \quad (3.6)$$

Vsaki spektralni komponenti želimo določiti koeficient stiskanja, ki se izračuna iz statistik σ_b , in sicer $\gamma_b = C(\sqrt{2}\sigma_b)^{\nu-1}$, kjer je ν parameter, ki določa količino beljenja, $C \approx 1,2$ pa je konstanta. Privzeta vrednost beljenja je $\nu = 0,33$, vendar so testi pokazali, da je včasih primernejša izbira $\nu = 0,1$, ki pomeni več beljenja. Stiskanje se opravi po celotnem spektru, toda koeficienti γ_b so določeni samo za osrednje frekvence filtrov, zato se manjkajoči

koeficienti izračunajo kot linearna interpolacija med dvema sosednjima osrednjima frekvencama. Manjkajoče koeficiente med frekvencama $f_{b,c}$ in $f_{b+1,c}$ geometrijsko interpretiramo kot točke na daljici s krajiščema $(f_{b,c} \frac{K_0}{f_s}, \gamma_b)$ in $(f_{b+1,c} \frac{K_0}{f_s}, \gamma_{b+1})$.

Zadnji korak spektralnega beljenja je inverzno filtriranje. Pobeljen spekter $|Y(k)|$ izračunamo tako, da koeficiente stiskanja pomnožimo z istoležnimi spektralnimi komponentami magnitudnega spektra $|X(k)|$, kar zapišemo kot $|Y(k)| = \gamma(k)|X(k)|$.



Slika 3.6: Učinek spektralnega beljenja.

Na sliki 3.6 je viden učinek spektralnega beljenja. Krivulji sta normalizirani tako, da je najvišja točka pri obeh enaka ena. V pobeljenem spektru so lokalni maksimumi bolj poudarjeni, vrhovi pa imajo relativno gledano med sabo manjšo višinsko razliko.

3.5 Iskanje F0

Računanje magnitudnega spektra s kratkočasovno Fourierjevo transformacijo je v raznih aplikacijah, ki časovno podan signal procesirajo v frekvenčni domeni, pogost pristop. Tudi iskanje osnovne frekvence v spektru je pogosto opravilo, vendar pa se algoritmi v tem delu med sabo ponavadi najbolj razlikujejo. Ideja Klapurijevega algoritma je preprosta in je bila podana že uvodoma — v magnitudnem spektru poiskati najbolj izstopajoče harmonično zaporedje, pri čemer je prvi ton zaporedja iskan ton. V uvodnem delu pričujočega poglavja so bile tudi na kratko predstavljene tri različne me-

tode obravnavanja polifoničnosti, vendar je implementirana samo iterativna metoda in zato edina izmed njih, ki je opisana v tem razdelku. V nadaljevanju je najprej opisana kriterijska funkcija (3.3) v diskretni domeni, temu sledi opis, kako so bile pridobljene uteži kriterijske funkcije, nazadnje pa je še podan učinkovit algoritem iterativnega ocenjevanja in odštevanja osnovne frekvence iz spektra.

3.5.1 Diskretna kriterijska funkcija

Ocenjevalna funkcija (3.3) je funkcija zvezne spremenljivke, ki predstavlja frekvenco v Hz. Najprej jo namesto frekvence f v Hz definiramo z zvezno periodo $\tau = \frac{f_s}{f}$ v enoti števila vzorcev:

$$s(T) = \sum_{m=1}^M w\left(\frac{f_s}{\tau}, m\right) |Y(m \frac{f_s}{\tau})|. \quad (3.7)$$

Ta oblika je povsem enaka prvotni (3.3), a z zamenjavo $f = \frac{f_s}{\tau}$. Če želimo magnitudni spekter $|Y(\cdot)|$ izračunati s pomočjo hitre Fourierjeve transformacije (FFT), ga prevedemo na diskretno obliko $Y[k]$, kjer je $k \in \mathbb{N}$. Dodatno uvedemo še parameter natančnosti $\Delta\tau$, ki omogoča izbiro ločljivosti, saj je poizvedovanje po $Y[k]$ omejeno samo na frekvence $\frac{kf_s}{K_0}$, $k = 0, 1, \dots, K_0 - 1$, medtem ko je τ zvezen parameter. Ocenjevalna funkcija (3.7), prilagojena za izračun s FFT, je

$$\hat{s}(\tau) = \sum_{m=1}^M w\left(\frac{f_s}{\tau}, m\right) \max_{k \in \kappa_{\tau, m}} |Y[k]|, \quad (3.8)$$

kjer $\kappa_{\tau, m}$ označuje množico zaporednih frekvenčnih košev m -tega harmoničnega delnega tona glede na osnovno frekvenco $\frac{f_s}{\tau}$ in vključuje tudi natančnost $\Delta\tau$. Naj $\langle \cdot \rangle$ označuje zaokroževanje k najbližjemu celemu številu. Potem je

množica $\kappa_{\tau,m}$ definirana kot

$$\kappa_{\tau,m} = \left[\left\langle \frac{mK_0}{\tau + \frac{\Delta\tau}{2}} \right\rangle, \left\langle \frac{mK_0}{\tau - \frac{\Delta\tau}{2}} \right\rangle \right] \cap \mathbb{N}. \quad (3.9)$$

Ko $\Delta\tau \rightarrow 0$, se množica $\kappa_{\tau,m}$ skrči na eno samo število, zato je v tem primeru max v (3.8) odveč, oblika ocenjevalne funkcije pa "prava" diskretna različica funkcije (3.7). Privzeta natančnost je $\Delta\tau = 0,1$. Zahvaljujoč hitremu algoritmu, opisanem v razdelku 3.5.3, izbrana vrednost $\Delta\tau$ nima večjega vpliva na računsko zahtevnost algoritma.

3.5.2 Utežitvena funkcija

Funkcija $w(\frac{f_s}{\tau}, m)$ iz prejšnjega razdelka 3.5.1 vzpostavi zvezo med osnovno frekvenco in višjimi harmoničnimi delnimi toni. Če opazujemo magnitude spektre posamičnih ravni tonov, zaigranih na določeno glasbilo, kjer se osredotočimo zgolj na osnovno frekvenco in njene harmonične delne tone, lahko opazimo določene odvisnosti med njimi. Delni toni so v spektru vidni kot lokalni vrhovi. Pri tem si lahko zastavimo različna vprašanja, kot npr. kako se vrhovi pripadajočih delnih tonov razlikujejo med seboj po velikosti, oz. kakšna so razmerja med njimi. Pri določenih glasbilih, zlasti pa pri petju, so v spektru vidna določena frekvenčna področja, ki imajo vedno nekoliko izrazitejše vrhove. Konkretno pri petju so ta področja dovolj izrazita, da imajo celo svoje ime – vokalni oblikovalci ali formanti. Naloga funkcije w je predstaviti ta razmerja.

Kadar imamo pri problemu transkripcije opravka z eno vrsto glasbil, je izbira w precej lažje opravilo kot v splošnem. Klapurijev algoritem je splošnonamenski, zato je za oceno w uporabil podatkovno zbirko 32 različnih glasbil. V svojem delu [8] je predstavil postopek simulacije, s katerim je optimiziral parametrizirano funkcijo $w(\frac{f_s}{\tau}, m)$. Zaradi poenostavitve procesa optimizacije in boljše zmožnosti posploševanja je to dvoparametrično funkcijo razstavil na produkt dveh funkcij $w(\frac{f_s}{\tau}, m) = w_1(\frac{f_s}{\tau})w_2(\frac{mf_s}{\tau})$ in optimiziral vsako posebej.

Učni podatki so zajemali različne mešanice tonov, skupaj z referenčnimi osnovnimi frekvencami. V postopku je bilo najprej naključno izbrano glasbilo, nato pa še naključen ton iz območja med 40 in 2100 Hz, ki se je primešal drugim tonom, dokler v mešanici ni bila dosežena zelena stopnja polifonije. Na ta način je bilo narejenih 4000 učnih primerov.

Stopnja polifonije je bila za vsako mešanico algoritmu podana kot parameter. Algoritem je analiziral 93 ms dolge časovne okne vhodnih podatkov, na izhod pa je podal seznam lokalnih maksimumov po funkciji $\hat{s}(\tau)$, skupaj s pripadajočimi osnovnimi frekvencami. Detekcija posamezne osnovne frekvence je štela kot napačna, če je od referenčne odstopala za več kot 3%. Minimizacijska kriterija sta bila dva: čim manjša povprečna napaka deleža napačnih detekcij v posameznem časovnem oknu in pa čim manjša povprečna napaka, kjer je kot napaka štel, če najmočnejša osnovna frekvenca po $\hat{s}(\tau)$ ni bila ena izmed tistih v referenci.

Optimizacija parametrov funkcije $w_2(\frac{mf_s}{\tau})$ je potekala na naslednji način. V frekvenčnem območju med 30 in 7000 Hz je bilo približno po geometrijski vrsti razporejenih 13 ključnih točk, ki so predstavljale parametre optimizacije in se jim je v postopku nastavljal amplitude. V primeru vmesnih frekvenčnih točk je bila uporabljena linearna interpolacija med obema sosednjima amplitudama. Postopno se je spreminjalo po eno amplitudo naenkrat. Za optimizacijo $w_1(\frac{f_s}{\tau})$ je bil uporabljen enak postopek, le ključnih točk je bilo 10 in frekvenčno območje 30 – 2500 Hz.

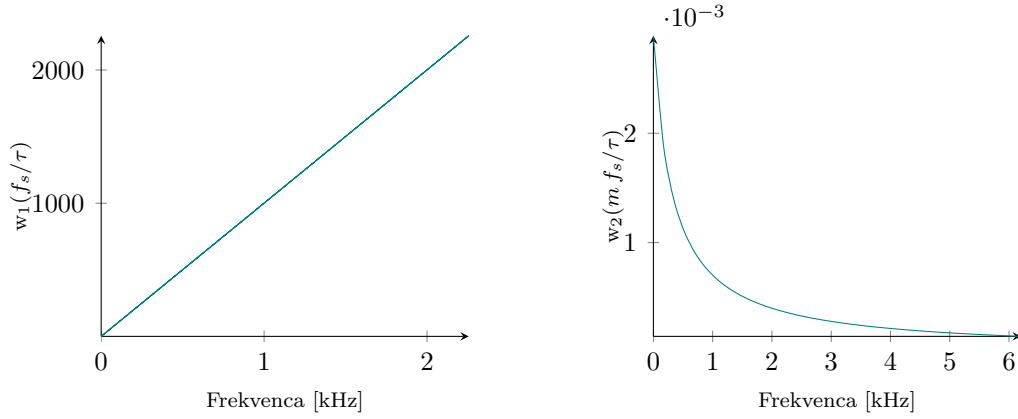
Dobljena funkcija w_1 je v linearni odvisnosti s frekvenco: $w_1(\frac{f_s}{\tau}) = f_s/\tau + \alpha$, funkcija w_2 pa je v recipročni odvisnosti od frekvence harmoničnega delnega tona: $w_2(\frac{mf_s}{\tau}) = 1/(mf_s/\tau + \beta)$. Če upoštevamo te rezultate in še konstante in parametre iz izvirne kode, se w_1 in w_2 izražata kot

$$w_1(\frac{f_s}{\tau}) = f_s/\tau + \alpha$$

$$w_2(\frac{mf_s}{\tau}) = \frac{\gamma}{24,7\rho + 0,1079 mf_s/\tau},$$

kjer so α , γ in ρ vhodni parametri algoritma. Slika 3.7 prikazuje obe utežitveni

funkciji.



Slika 3.7: Levi graf prikazuje utežitveno funkcijo $w_1(\frac{f_s}{\tau}) = \frac{f_s}{\tau} + 52$, desni pa funkcijo $w_2(\frac{m f_s}{\tau}) = \frac{0,1}{34,58 + 0,1079 m f_s/\tau}$ pri parametrih $\alpha = 52$, $\gamma = 0,1$ in $\rho = 1,4$.

Linearna funkcija pokriva zgolj frekvenčno območje, kjer se nahajajo kandidati za osnovne frekvence, medtem ko druga funkcija pokriva celotno frekvenčno območje. Faktorizacija funkcije w na dve funkciji omogoča enostavno implementacijo. Magnitudni spekter je mogoče najprej utežiti z w_2 vzdolž celotnega frekvenčnega območja, nato lahko $\hat{s}(\tau)$ izračunamo brez uteži, nazadnje pa, da ne favoriziramo zgolj nizkih frekvenc, pred iskanjem maksimuma izmed kandidatov za F_0 , vrednosti $\hat{s}\tau$ pomnožimo z $w_1(\frac{f_s}{\tau})$.

Da nizke frekvence res niso favorizirane, lahko ponazorimo z naslednjim primerom. Naj bo magnitudni spekter enak ena po celem frekvenčnem območju. Naj bosta f_0 in f_1 , izražena v Hz in $f_0 \neq f_1$, dva kandidata za F_0 . Funkcijo w_1 poenostavimo v $w_1(f) = f$ in w_2 poenostavimo v $w_2(m f) = \frac{1}{m f}$. Ko magnitudni spekter utežimo z w_2 , vrednosti kriterijske funkcije pri kandidatih f_0 in f_1 zaporedoma znašata

$$s(f_0) = \frac{1}{f_0} + \frac{1}{2f_0} + \dots + \frac{1}{M f_0},$$

$$s(f_1) = \frac{1}{f_1} + \frac{1}{2f_1} + \dots + \frac{1}{M f_1}.$$

Iz $f_0 \neq f_1$ sledi, da $s(f_0) \neq s(f_1)$. Ko pa ju pomnožimo s funkcijo w_1 , je

$$\begin{aligned} s(f_0)w_1(f_0) &= f_0 \left(\frac{1}{f_0} + \frac{1}{2f_0} + \dots + \frac{1}{Mf_0} \right) \\ &= 1 + \frac{1}{2} + \dots + \frac{1}{M} \\ &= f_1 \left(\frac{1}{f_1} + \frac{1}{2f_1} + \dots + \frac{1}{Mf_1} \right) \\ &= s(f_1)w_1(f_1). \end{aligned}$$

△

3.5.3 Iterativno ocenjevanje in odštevanje

Pri iskanju optimalnih parametrov utežitvene funkcije je bila uporabljena t.i. direktna metoda, ki v magnitudnem spektru z uporabo kriterijske funkcije $\hat{s}(\tau)$ poišče fiksno število najboljših kandidatov za F0. Prvi problem tega pristopa je, da je potrebno $\hat{s}(\tau)$ ovrednotiti za vse periode τ , ciljna frekvenčna ločljivost pa je lahko manjša, kot jo imajo FFT koši, zato časovna zahtevnost hitro naraste. Drugi problem je nerobustna detekcija, četudi je sama kriterijska funkcija dokaj robustna. Težava nastane, kadar je kakšen ton bolj izrazit in ima posledično običajno tudi visoke harmonične delne tone, npr. drugega, ki je z F0 v oktavnem razmerju, ali tretjega, ki je duodecima (tj. oktava in kvinta) nad F0.

Drugo težavo lahko do neke mere odpravimo tako, da prepoznan ton, skupaj z vsemi harmoničnimi delnimi toni, odštejemo iz spektra in nato postopek iskanja in odštevanja ponavljamo, dokler v spektru ostane dovolj energije oz. ne dosežemo želene stopnje polifonije. Po vsaki ponovitvi iskanja najboljšega kandidata in odštevanju le tega je, kot da bi algoritem začel znova. Zato zadostuje, da s $\hat{s}(\tau)$ poiščemo samo tisto periodo τ , kjer je dosežen maksimum. Izkaže se, da lahko z razdelitvijo frekvenčnega območja na bloke in uporabo ustrezne optimistične heuristike maksimum učinkovito poiščemo po metodi najprej najboljši, s tem pa rešimo še prvi problem. V nadaljevanju sta predstavljena oba postopka.

Algoritem 3.5.1 Iterativno prepoznavanje in odštevanje.

Vhod: $Y, win, F, P, w, K_0, f_s$

Izhod: $F0s, Sals$

```

1  $Y_R \leftarrow |Y|$ 
2  $W \leftarrow |\text{FFT}(win)|$ 
3  $\ell \leftarrow \lfloor \text{length}(W)/2 \rfloor$ 
4 for  $n = 1$  to  $F$  do
5    $p \leftarrow 0$ 
6    $c \leftarrow 1$ 
7    $F0s(n) \leftarrow [0]$ 
8    $Sals(n) \leftarrow [0]$ 
9    $\tau_p \leftarrow 0$ 
10  while  $p < P$  do
11     $\hat{\tau} \leftarrow \arg \max_{\tau} \hat{s}(\tau)$ 
12    if  $\hat{\tau} \notin F0s(n)$  then
13       $c \leftarrow 1$ 
14       $p \leftarrow p + 1$ 
15       $F0s(n, p) \leftarrow f_s / \hat{\tau}$ 
16       $Sals(n, p) \leftarrow s(\hat{\tau})$ 
17    end if
18     $Y_D \leftarrow [0]$ 
19    for  $m = 1$  to  $M$  do
20       $Y_D(\langle \frac{m K_0}{\hat{\tau}} \rangle - \ell : \langle \frac{m K_0}{\hat{\tau}} \rangle + \ell) \leftarrow w(\frac{m f_s}{\hat{\tau}}) W(1 : 2 \cdot \ell + 1)$ 
21    end for
22     $Y_R(n) \leftarrow Y_R(n) - \max\{0, c Y_D |Y|(n)\}$ 
23    if  $\tau_p \neq \hat{\tau}$  then
24       $c \leftarrow 2c$ 
25    else
26       $c \leftarrow 1$ 
27    end if
28     $\tau_p \leftarrow \hat{\tau}$ 
29  end while
30 end for

```

Psevdokoda 3.5.1 prikazuje postopek iterativnega prepoznavanja in odštevanja. Vhodni parametri algoritma so frekvenčni spekter Y za vsak časovni okvir, v časovni domeni izražena okenska funkcija win , število časovnih okvirjev F , največja stopnja polifonije P , utežitvena funkcija w , velikost FFT okna K_0 in frekvenca vzorčenja f_s . Algoritem za vsak časovni okvir izračuna P osnovnih frekvenc (v $F0s$), skupaj s pripadajočo energijo (v Sals).

Koraki iterativnega algoritma so naslednji:

1. V Y_R se nastavi vhodni magnitudni spekter in tekom izvajanja vsebuje preostanek magnitudnega spektra, ki je rezultat odštevanja zvokov iz njega. V W se pripravi glavni reženj FFT časovnega okna, ki se uporabi za odštevanje.
2. Za vsak časovni okvir izmed F -tih se poišče P različnih najboljših kandidatov za F0. Najboljši kandidat, izražen s periodo $\hat{\tau}$, se v vsaki ponovitvi izračuna s hitrim postopkom po metodi najprej najboljši (opisan je kasneje) iz preostanka magnitudnega spektra Y_R .
3. Če je najboljši kandidat $\hat{\tau}$ še nedetektiran ton v trenutnem časovnem okvirju, se ga doda v $F0s(n)$, njegovo energijo pa v $Sals(n)$.
4. V Y_D se v frekvenčne koše $\langle \frac{mK_0}{\hat{\tau}} \rangle$ pri $\{m\}_1^M$ in njihovo lokalno okolico na podlagi spektralne vsebine okenske funkcije W pripravi vsebino za odštevanje iz Y_R , ki je primerno utežena z utežitveno funkcijo w .
5. Iz Y_R se odšteje zaznan zvok v Y_D .
6. Postopek se za vsak časovni okvir ponavlja, dokler ni dosežena največja stopnja polifonije, P .

Spremenljivka c nadzira velikostni red odštevanja detektiranega zvoka. V primeru, da se v preostanku magnitudnega spektra istega časovnega okvirja isti ton $\hat{\tau}$ pojavi večkrat, se ga v izhodno strukturo $F0s$ doda samo prvič, iz spektra pa se ga odšteva vsakič. Če se isti ton ponavlja zaporedoma, se velikostni red odštevanja podvoji, da se proces pohitri. Vendar pa iz spektra

ni dobro odšteti preveč agresivno, da ne pokvarimo nadaljnjih detekcij. Za višje harmonične delne tone za to poskrbi utežitev w , pri nižjih pa ima c večji vpliv.

Dve periodi τ_1 in τ_2 pripadata istemu tonu, če sta dovolj blizu skupaj, npr. če nista več kot četrta tona narazen, kar lahko ob upoštevanju temperirane uglasitve izrazimo kot $|12 \log_2(\tau_1/\tau_2)| \leq 0,5$.

Pri računanju Y_D je potrebno upoštevati ustrezno okensko funkcijo — ti-
sto, s katero je bil izračunan frekvenčni spekter časovnega signala. Ni dovolj, da se odšteje zgolj vrhove v frekvenčnih koših $\langle \frac{mK_0}{\tau} \rangle$ za $\{m\}_1^M$, pač pa je potrebno upoštevati, da se spektralna vsebina okrog osrednje frekvence razlije še v nekaj sosednih košev. Razlog za to je stranski učinek rezanja časovnega signala z oknom, saj tak odrezan signal ni periodičen, kar v frekvenčni domeni povzroči odzvanjanje stran od osrednje frekvence. Nepravokotne okenske funkcije, ki ob robovih stisnejo amplitudno ovojnico signala v nič, učinek odzvanjanja zmanjšajo, povečajo pa širino glavnega režnja spektra.

3.5.4 Hitro iskanje najboljšega kandidata

Čeprav odzvanjanje spektra povzroča določene težave pri odštevanju, pa jih toliko manj pri detekciji vrhov, saj lahko veliko energije zajamemo že, če se osrednji frekvenci zgolj dovolj približamo. Če to upoštevamo ob dejstvu, da v posamezni iteraciji prepoznavanja in odštevanja zadostuje en sam najboljši kandidat, lahko hitro iskanje maksimuma funkcije $\hat{s}(\tau)$ izvedemo na sledeč način:

1. Algoritem išče maksimum funkcije $\hat{s}(\tau)$ za periode $\tau \in [\tau_{\min}, \tau_{\max}]$ z natančnostjo iskanja τ_{prec} .
2. Frekvenčno območje iskanja F0, ki ga v frekvenčnih koših izrazimo kot interval $[K_0/\tau_{\max}, K_0/\tau_{\min}]$, razdeli na B neprekrivajočih se blokov. Vsak blok b na začetku zajema periode med $\tau_{\min} + \frac{b-1}{B}(\tau_{\max} - \tau_{\min})$ in $\tau_{\min} + \frac{b}{B}(\tau_{\max} - \tau_{\min})$.

3. Označimo spodnjo in zgornjo mejo bloka b zaporedoma kot τ_b^\perp in τ_b^\top , ki zajema frekvenčne koše med K_0/τ_b^\top in K_0/τ_b^\perp . Algoritem za vsak posamezni blok b izračuna zgornjo mejo za $\hat{s}(\tau)$. To stori tako, da na vsakem frekvenčnem intervalu, katerega krajišči sta $k_1(m) = \langle m K_0/\tau_b^\top \rangle$ in $k_2(m) = \langle m K_0/\tau_b^\perp \rangle$, izračuna maksimum v magnitudnem spektru Y_R , ga uteži z vrednostjo funkcije w_2 pri spodnji meji tega intervala, in dobljene maksimume sešteje za vse m . Vsoto nato uteži še z $w_1(\langle K_0/\tau_b^\perp \rangle)$:

$$\forall \tau : \tau_b^\perp \leq \tau \leq \tau_b^\top : \hat{s}(\tau) \leq w_1(\langle K_0/\tau_b^\perp \rangle) \cdot \sum_{m=1}^M w_2(\langle m K_0/\tau_b^\top \rangle) \max\{Y_R(k_1(m)), \dots, Y_R(k_2(m))\}. \quad (3.10)$$

4. Poišče blok b^* , kjer je izmed vseh blokov dosežena največja vrednost (3.10) in ga razdeli na dva bloka enakih dolžin.
5. Za oba nova bloka ponovno izračuna (3.10).
6. Koraka 4 in 5 ponavlja, dokler v najboljšem bloku širina frekvenčnega intervala ni manjša od τ_{prec} .
7. Rezultat je perioda $\frac{1}{2}(\tau_{b^*}^\perp + \tau_{b^*}^\top)$ in zadnja izračunana energija najboljšega bloka.

Prepričajmo se, da hitro iskanje res najde najboljšega kandidata $\hat{\tau}$. Opazujemo vsoto (3.10). Na vsakem posameznem frekvenčnem intervalu m -tega harmoničnega delnega tona, tj. interval $I_m = [\langle m K_0/\tau_b^\top \rangle, \langle m K_0/\tau_b^\perp \rangle]$ poiščemo maksimum magnitudnega spektra na tem intervalu. Očitno nobena druga vrednost ni višja. Ta maksimalna vrednost je utežena z monotono padajočo funkcijo w_2 na levem krajišču intervala I_m , kjer je na tem območju funkcija w_2 največja. Torej na posameznem intervalu I_m vedno dobimo res največjo vrednost. Vse omenjene lastnosti veljajo tudi za preostale intervale, zato je vsota dobljenih maksimumov res zgornja meja. Uteževanje vsote z monotonno naraščajočo funkcijo w_1 z vrednostjo, ki jo doseže v desnem frekvenčnem krajišču bloka, pomeni množenje z najvišjo možno vrednostjo w_1

na tem območju. Od tod sledi, da razpolovitev bloka ne more povečati zgornje meje energije za nobenega od novih dveh blokov. \square

Iskanje se lahko začne z enim samim blokom ($B = 1$). Klapuri začne z $B = \sqrt{\frac{\tau_{\max} - \tau_{\min}}{\tau_{\text{prec}}}}$ bloki.

3.5.5 Določanje stopnje polifonije

Iterativno prepoznavanje in odštevanje ne poskuša ugotoviti stopnje polifonije. Število osnovnih frekvenc in njihove energije, ki jih vrne za vsak okvir, je vedno P . V splošnem se pri polifoničnih posnetkih število glasov ali inštrumentov tekom časa spreminja, zato je dobro algoritem nekako dopolniti, da poskuša ta problem rešiti npr. na osnovi energije, ki jo izračuna za vsak vrnjen ton.

Kadar ima zvočni posnetek malo dinamičnih razlik, je morda dovolj dobra ocena za polifonijo kar fiksno upragovljenje. V tem primeru tone, ki imajo energijo manjšo od nekega praga, zavržemo. Klapuri predlaga ustavitveni kriterij, ki za novo razpoznan ton $\hat{\tau}_p$ v ponovitvi p ne poveča več količine

$$S(p) = \frac{\sum_{i=1}^p \hat{s}(\hat{\tau}_i)}{p^\phi}, \quad (3.11)$$

kjer je vrednost $\phi = 0,7$ bila pridobljena izkustveno. Pri $\phi = 1$ dobimo povprečno vrednost energije, vrednost vsote $S(p)$ pa z vsakim nadaljnjim p pada. V tem primeru bi algoritem zajel enoglasje. Pri $\phi = 0$ dobimo vsoto energij, ki pa vedno narašča, torej bi algoritem vrnil vseh P glasov. Iz tega razloga ti dve mejni vrednosti v splošnem ne delujeta.

Poglavje 4

Priprava podatkov

Velik del problema je pogosto potrebno rešiti že s pripravo podatkov. V tem poglavju si pogledamo, za kakšne podatke gre, kako so bili pridobljeni, kako se jih je pripravilo v ustrezno obliko in kakšne so bile težave pri tem.

4.1 Opis podatkovne množice

Podatki, ki so predmet avtomatične transkripcije, so del terenskih zvočnih posnetkov, zajetih iz različnih krajev po Sloveniji. Njihova vsebina poskuša zajeti ljudske običaje. Med njimi najdemo govore, deklamacije, igranje inštrumentov in ubrano ljudsko petje. Posnetki so del zbirke ZRC SAZU. Na osnovi teh posnetkov so izdali tudi knjige. Zbirka slovenskih ljudskih pesmi, ki nosi istoimenski naslov [7], vsebuje po pokrajinah označene pesmi, vsaka izmed njih je v knjigi zapisana z notacijo ene kitice, skupaj z besedilom vseh kitic. Zraven je pripisano še krajinsko področje. Pesmi so urejene po tematikah.

4.1.1 Priprava pesmi

V terenskih posnetkih je bilo potrebno poiskati večglasne péte pesmi. Iskanje po posnetkih je potekalo ročno. V evidenčni seznam (kot datoteka) se je k imenu terenskega posnetka označilo časovna mesta začetkov in koncev delov,

ki so vsebovali večglasno petje. Ko je bilo prečesavanje posnetkov končano, se je napisalo skripto, ki je iz dolgih terenskih posnetkov izluščila želene pesmi. Za rezanje posnetkov je bil uporabljen program sox¹.

Poleg celotnih dolgih terenskih zvočnih posnetkov so bila za vsako pesem na voljo tudi besedila in bolj ali manj dobre transkripcije ene kitice (tisto, kar je vključeno v prej omenjeno knjigo Slovenske ljudske pesmi). Slednje so bile na voljo kot Sibeliusove² datoteke (program za notacijo), nekatere pa že kot MIDI datoteke. Manjkajoče MIDI datoteke se je pridobilo z izvozom iz Sibeliusa.

4.1.2 Seznam pesmi

Iz terenskih posnetkov je bilo pridobljenih 85 pesmi. Od teh je bilo zaradi obilice ročnega dela (tj. transkripcije) v končno uporabno obliko pripravljenih le 37. Pesmi so izključno večglasno petje. Število glasov je med dva in štiri, število pevcev in pevk pa je lahko tudi večje. Glasovi so različni: lahko so samo moški, samo ženski ali mešani.

Tabela 4.1 vsebuje seznam pesmi in njihovo polifoničnost (število različnih glasov). Tematika pesmi je žalostna, veliko naslovov je ponovljenih. Gre za različice pesmi po okrajih. Včasih to poleg nekoliko drugačnega narečja in besedila pomeni tudi drugačno melodijo.

4.2 Samodejna poravnava

Preden je bila sprejeta odločitev o ročni transkripciji posnetkov, je bil narejen poskus samodejne poravnave zvočnega posnetka z MIDI datoteko. Vseeno pa je bilo kot predpripravo potrebno postoriti nekaj ročnega dela. Obstoječo transkripcijo prve kitice je bilo potrebno prestaviti v pravilno tonaliteto in popraviti morebitne napačne višine tonov. Nato je bilo potrebno to kitico kopirati tolikokrat, kolikor je kitic v zvočnem posnetku. Nazadnje se je pognalo

¹<http://sox.sourceforge.net>

²<http://www.sibelius.com>

algoritem za poravnavo.

Preizkušena sta bila dva algoritma, prvi je delo Ellisa in Turetskega [19], ki poravnava zvočne posnetke in MIDI, drugi pa je MATCH avtorjev Dixona in Widmerja [3], ki podpira zgolj ujemanje dveh zvočnih posnetkov.

Prvi deluje tako, da tone v MIDI datoteki predstavi kot vrhove v ustreznih frekvenčnih koših spektrograma. Ta se nato primerja s spektrogramom zvočnega posnetka. Neka kriterijska funkcija določi ujemanje med posameznima spektrogramoma. Zgradi se matrika vseh možnih parov spektrogramov. Nato se izvede časovno ukrivljanje z uporabo dinamičnega programiranja. Pri tem se omeji začetno in končno točko, zato je pri uspešni poravnavi pričakovano, da pot od začetne točke do končne poteka bolj ali manj po glavni diagonalni (ali antidiagonalni, odvisno od položaja omenjenih točk).

Rezultati poravnave s prvim algoritmom so obetavni, dokler opazujemo najboljše ujemačo se pot na diagonalni matrike. Slušno rezultati niso bili tako dobri. V glavnem je algoritem praviloma uspel slediti zvočnemu posnetku, vendar z nekaterimi pretiranimi zadržki in pohitritvami. Poleg tega takšen MIDI nima informacije o dobah in tempu, saj predpostavi fiksno tempo, recimo 60 udarcev na minuto, trajanja tonov pa so prilagojena časovnemu ujemanju, zato informacije o dobah ni več.

Drugi algoritem, MATCH, pa je dal še slabše rezultate. Pravzaprav ni našel nobene korelacije med MIDI in zvočno datoteko. Za nameček je bilo potrebno MIDI datoteke pred poravnavo še sintetizirati v zvok.

4.3 Ročna transkripcija

Slabi rezultati poskusa avtomatične poravnave, napačno določene tonalitete v obstoječem notnem zapisu in raznolikosti posameznih kitic so vodili do odločitve, da bo transkripcijo najbolje opraviti ročno. To se je po vseh pričakovanih izkazalo za naporno opravilo, ki je vzelo precej časa. Zlasti težavni so bili tisti posnetki, kjer nastopajo vsaj trije glasovi, še posebej pa kadar posamezen glas pôje več pevcev. V nadaljevanju je opisan sam potek

ročne transkripcije. Zaradi možnosti urejanja tako zvočnega kot MIDI zapisa in podpore enostavnemu določanju tempa je bila uporaba orodja Steinberg Cubase³ logična izbira.

4.3.1 Cubase

Digitalna zvočna delovna postaja (*Digital Audio Workstation*) Cubase je profesionalna programska oprema, namenjena studijskemu snemanju in digitalni obdelavi zvoka tako v realnem času kot tudi že obstoječih posnetkov. Podpira tudi MIDI datoteke, katere je mogoče ustvariti in urejati na več načinov (kot npr. note ali seznam dogodkov). Namenjen je širokemu občinstvu: skladateljem, producentom, glasbenim skupinam, pevcem, kitaristom ter producentom elektronske in ritmične glasbe [22]. V nadaljevanju na kratko opišemo tiste dele njegovega grafičnega uporabniškega vmesnika, ki so bili za lažjo izdelavo transkripcij ključni.

Ko ustvarimo nov ali odpremo obstoječ projekt, se nam prikaže delovna površina, kjer so navpično nanizane sledi, ki se raztezajo v vodoravni smeri (ali pa te sledi šele ustvarimo). Sledi so lahko različnih tipov, vsak tip pa določa vrsto prikazane vsebine, kot npr. MIDI sled, sled z zvočnim zapisom, sled z besedilom, sled s tempom, Tako je material različne vrste ločen med seboj.

Nad delovno površino je merilo, ki lahko prikazuje različne časovne enote: čas v sekundah, minutah in urah, dobe, vzorce idr. Pri poravnavi tempa in dob je to merilo zelo uporabno. Pri postavljanju taktnic se na merilu pojavijo rumeni navzdol obrnjeni trikotniki, vidne pa so tudi navpične črte nad zvočnim posnetkom. Opozorimo, da taktnica ni povsem ustrezen izraz, saj je črte mogoče postavljati kjerkoli, odvisno od nastavitve kvantizacije in lepljenja (*snap*, bomo pa vseeno ta izraz uporabljali v nadaljevanju).

Seveda ima tudi sama vsebina sledi določeno grafično podobo. Zvočni posnetki so prikazani kot amplitudne ovojnice v krivuljni obliki. Višine in dolžine tonov MIDI datotek so prikazani kot pravokotniki v mreži. Ob

³<http://www.steinberg.net>

strani je navpično prikazana klaviatura. Višina tona je določena z navpičnim položajem pravokotnika glede na tipkovnico, dolžine pravokotnikov pa ustrezajo dolžini tonov, vendar relativno glede na izbrano metriko v merilu.

Višine in dolžine MIDI tonov je mogoče urejati z orodji kot so svinčnik, radirka in škarje. S svinčnikom lahko narišemo nove pravokotnike, z radirko jih brišemo, s škarjami režemo. Z miškinim kazalcem jih lahko premikamo in ob robovih tudi spreminjamo dolžino.

Cubase vodi seznam vseh odprtih sledi in vseh zvočnih posnetkov, ki jih uporablja v ozadju. Seznam je dostopen preko glavnega menija in se imenuje *Media pool*. Kadar projektu želimo nastaviti tempo glede na tempo obstoječega zvočnega posnetka, je pomembno, da v tem seznamu pri zvočnih posnetkih izključimo t.i. glasbeni način (*Music mode*), saj v nasprotnem primeru spreminjanje tempa projekta povzroči spremembo tempa zvočnega posnetka, kar ni želen učinek.

Za prilagajanje tempa projekta zvočnemu posnetku je zelo uporabno orodje *Warp tool*, ki omogoča postavljanje in premikanje taktnic na ustrezna mesta. Pri tem je pomembno, da to počnemo od začetka proti koncu in ne od poljubnega mesta, sicer moramo včasih razveljaviti celotno zaporedje napačno postavljenih taktnic, vse do mesta napake.

Ena izmed tipov sledi je tudi tempo sled, ki spremembe tempa prikazuje grafično. Med poljubnima dvema taktnicama je mogoče določiti postopno spremembo tempa (*Linear mode*) ali pa je sprememba nenadna (*Step mode*). Ko izvažamo posnetke, je tempo sled mogoče izključiti. V tem primeru se tempo projekta ne izvozi. Slednje je lahko uporabno, kadar si pripravljamo transkripcije, ne želimo pa podati informacije o tempu, npr. za algoritmično poravnavo med zvokom in MIDI posnetkom.

4.3.2 Časovna poravnava

Cilj časovne poravnave je časovno ujemanje zapisa v MIDI datoteki z ustreznim časovnim mestom v zvočni datoteki. Zvočni posnetek se je najprej uvozilo v Cubase. Določiti je bilo potrebno taktovski način prvega takta in

oceniti začetni tempo posamezne skladbe. Pri tem so bile nekoliko v pomoč obstoječe transkripcije. Zvočni posnetek se je zamaknilo v desno, tako da sta se prvi dobi posnetka in sledi s tempom ujemali. Nato je sledila poravnava. Zaradi nenehnega nihanja v tempu je bila poravnava večinoma potrebna po vsaki dobi, včasih pa celo po ustrezni poddelitvi. Pri nekaterih posnetkih je bilo potrebno paziti še na menjave taktovskega načina. Ritmična poravnava je bila lažji del procesa transkripcije.

4.3.3 Določitev tonskih višin

Dosti težje opravilo je bila transkripcija melodije, basa in vseh ostalih glasov. Običajno se je v Cubase v novo sled uvozilo MIDI datoteko z obstoječo eno kitico transkripcije, v drugih primerih pa se je MIDI napisalo od začetka. Predhodna ritmična poravnava zvočnega posnetka oz. določitev tempa projekta je pri uvozu samodejno prilagodila tempo MIDI zapisa in dolžine posameznih tonov, tako da so se ujemali z zvočnim posnetkom. Tako višine kot točne dolžine posameznih tonov je bilo potrebno še ročno popraviti, včasih tudi določiti pravilno tonaliteto. Na nekaterih mestih je bilo potrebno izboljšati časovno poravnavo z ustrezno poddelitvijo. Urejeno kitico se je potem kopiralo na ustrezno mesto naslednje kitice v zvočnem zapisu ter prilagodilo na enak način in tako za vse kitice do konca.

4.3.4 Izvoz

Končni rezultat ročne transkripcije sta bili dve datoteki za vsako skladbo, ena MIDI, ki vsebuje informacijo o notnih višinah in dolžinah, dobah, taktovskem načinu in času pojavitve in konca vsakega tona, ter druga, originalni zvočni posnetek z nekoliko dodane tišine na začetku in podvzorčenjem.

Ko je bila transkripcija končana, se je izvozilo obe sledi, tako MIDI datoteko kot tudi zvočni posnetek, saj je bilo na začetku zvočnega posnetka zaradi lažje poravnave dodane nekoliko tišine. Cubase je pri izvozu v MIDI spremembe v tempu in taktovskem načinu shranil v posebno konduktersko

sled. MIDI tako ohrani informacijo o dobah, vendar pa je pri kasnejši obdelavi za pravilno časovno interpretacijo zvočnih dogodkov potrebno upoštevati podatke z omenjene sledi. To je posebej izpostavljeno, saj knjižnica MIDI Toolbox [5] pri uvozu MIDI datoteke v MATLAB tovrstnih podatkov ne upošteva in je zato bilo potrebno uvažanje ustrezno dopolniti z lastnim programom, sicer bi bila časovna poravnava brezpredmetna.

Zvočni posnetek se je izvozilo v MP3 format. Morebitni dva kanala (*stereo*) se je združilo v enega samega (*mono*). Originalni posnetki so vzorčeni s frekvenco 48 kHz. Ker Klapurijev algoritem deluje samo pri 44,1 kHz, je bilo opravljeno tudi podvzorčenje.

4.3.5 Težave

Pri ročni transkripciji so bile določene težave. Izvor le-teh najdemo tako pri izvajalcih, kvaliteti zajemanja posnetka, kot tudi pri transkripciji sami, brez prej naštetih problemov.

4.3.6 Kakovost zajemanja posnetkov

Kot smo že omenili, gre za terenske posnetke. Nadzor nad kvaliteto je omejen in slabši od studijske. Čas pridobivanja teh posnetkov sega lahko tudi globoko v 20. stoletje, tj. v čas izključno analogne opreme za zajemanje. V modernejšem času so bili zapisi pretvorjeni v digitalnega. Kot posledica teh in še nekaterih drugih vplivov so nekateri posnetki šumni, komaj slišni, vendar je velika večina, s katero smo imeli opravka, solidne kakovosti, včasih celo presenetljivo dobre glede na vse okoliščine. Domnevno se je pevce in pevke snemalo z enim mikrofonom, zato so jakosti glasov včasih neizenačene, mestoma pa se ta razmerja tudi kar občutno spreminjajo. Dobra lastnost posnetkov pa je, da so suhi, torej vsaj nimamo opravka z odmevi.

4.3.7 Kakovost izvajanja pesmi

Izpostaviti je potrebno tudi problem izvedbe. Ker gre za ljudsko petje, so pevci in pevke večinoma netrenirani, zato sama vokalna produkcija ni na najvišjem nivoju. Po barvi glasu sodeč gre večinoma za starejše osebe. Nemalokrat se zgodi, še zlasti kadar je veliko kitic, da se nekateri izvajalci ne spomnijo pravega besedila, zato nekako capljajo za glavnim, vodilnim pevcem (ali pevko), imajo svojo različico besedila, ali pa momljajo nerazložno besedilo. To je lahko večji problem kot se zdi na prvi pogled, saj pri neizraziti izgovorjavi ni jasnih, odprtih vokalov, podira pa se tudi ritmična struktura petja. So tudi primeri, kjer se pevci tako zapletejo, da se ustavijo in začnejo kitico peti ponovno.

Pogosto je vprašljiva tudi intonacija, to je višina tonov, tako posameznih glasov v razmerju do drugih, kot tudi vseh glasov relativno na komorno uglastitev (440 Hz za ton a1). Običajno se petje, govorimo o tonaliteti (tj. relativni višini glede na komorni ton), prične v intonaciji. Najbrž jim je intonacijo dal snemalec. Ščasoma pa intonacija odtava, najpogosteje so tonaliteto višali. Pri ročni transkripciji se je pogosto pojavila dilema, kako določiti tonaliteto, saj so peli nekje vmes med dvema možnima tonalitetama. MIDI pač ne omogoča četrta tonov.

Še ena značilnost so neodločni začetki, oz. pogosteje nestabilni, saj intonacija ponavadi vznikne iz neke prenizke in po nekem kratkem času preide do prave. Na kratko lahko ta pojav imenujemo drsenje. Te je v MIDI datoteko težko opisati zgolj z višinami; nekoliko lažje bi jih bilo opisati kot učinek, ki pa jih nismo uporabljali.

4.3.8 Pomoč pri transkripciji

Transkripcija sama pa je težavna, čim je opravka s tremi ali več glasovi, po možnosti še s podvojenimi posameznimi glasovi. Pri tem si pomagamo na različne načine. Posnetek lahko z ustreznim programom, kot je Audacity⁴

⁴<http://audacity.sourceforge.net/>

(ali tudi Cubase), predvajamo počasneje, brez spremembe višine. Pomagamo si lahko tudi tako, da izberemo kratek problematičen segment in ga vrtimo v zanki, da zveni kot dolg ton. Včasih lahko segmentu tudi spremenimo višino, da je problematičen glas v za človeško uho bolj prijazni legi. Uporaba glasbila in znanje harmonije in muzikologije v splošnem pa sta vedno dobrodošla.

Št.	Naslov	Št. glas.
2	Mačeha in sirota	3
3	Oče umori sinova	3
4	Vdovec na ženinem grobu	3
5	Nevesta detomorilka	3
6	Obsojena detomorilka	2
7	Mačeha in sirota	3
8	Žalostna usoda treh sinov	3
9	Nevesta detomorilka	2-3
10	Nezvesta gospa s tremi stražarji	2
11	Zapuščene sirote	2
12	Mačeha in sirota	2
13	Mačeha in sirota	2
14	Žalostna usoda treh sinov	2
15	Nevesta detomorilka	2
16	Obsojena detomorilka	2
17	Mačeha in sirota	2
18	Nezvesta gospa s tremi stražarji	2
19	Zapuščene sirote	3
20	Nevesta detomorilka	3
21	Mačeha in sirota	2
22	Žalostna usoda treh sinov	3
23	Mačeha in sirota	2
24	Vdovec na ženinem grobu	3
25	Oče umori sinova	2
26	Vdovec na ženinem grobu	3
27	Nezvesta gospa s tremi stražarji	4
28	Vdovec na ženinem grobu	2-3
29	Obsojena detomorilka	4
30	Mačeha in sirota	3-4
31	Obsojena detomorilka	3
32	Žalostna usoda treh sinov	2
33	Nezvesta gospa s tremi stražarji	2
34	Nezvesta gospa s tremi stražarji	2
35	Vdovec na ženinem grobu	2
36	Nevesta detomorilka	2
37	Oče umori sinova	2
38	Brat in sestra se najdeta	4

Tabela 4.1: Zbirka večglasnih ljudskih pesmi.

Poglavje 5

Interaktivna vizualizacija

Čeprav je uho, tako kot oko, eno izmed naših čutil, si ljudje stvari pogosto radi predstavljamo v grafični podobi. Kako bi si prikazali glasbo? Ob poslušanju zvočnega posnetka si lahko predstavljamo skupino glasbenikov ali kar cel orkester, ki igra. Lahko si v mislih prikažemo glasbila. Glasbeniki si včasih med izvajanjem skušajo priklicati note v spomin. Za boljšo izvedbo si lahko predstavljajo širono občinstvo in veličastne dvorane.

Tukaj želimo doseči nekaj podobnega. Delovanje algoritma je lažje razumeti, če si ga lahko predstavimo grafično. Klapurijev algoritem večino dela opravi v magnitudnem spektru, zato je smiselno, če si ga na nek primeren način prikažemo. Ena izmed primernih vizualizacij se zdi prikaz iterativne detekcije in odštevanja po posameznih korakih, kjer bi se dalo videti, kateri ton je sistem prepoznal, kako ga je umestil v spekter, kakšen je preostanek spektra po odštevanju prepoznanega tona ter kaj se s spektrom dogaja v nadaljevanju.

Razvit je bil grafični vmesnik, podoben opisu zgoraj, z nekaj še dodatne funkcionalnosti. Opišemo ga v tem poglavju. Uporabimo ga na praktičnem primeru analize napak pri detekciji tonov.

5.1 Opis aplikacije

Aplikacija za vizualizacijo je bila napisana v MATLABU. Po zagonu aplikacija prebere zvočni in referenčni MIDI posnetek, nato izvede Klapurijev algoritem. Dobljene rezultate ovrednoti. Temu sledi izris grafične podobe.

Grafični uporabniški vmesnik je sestavljen iz dveh delov. Zgornji del prikazuje zvočni signal kot valovno krivuljo in razdelitev signala na časovno prekrivajoče se okvirje. Spodnji del prikazuje magnitudni spekter trenutno izbranega časovnega okvirja. Po obeh delih je mogoče z miško prosto navigirati v levo in desno in s kolesčkom spreminjati povečavo.

5.1.1 Prikaz časovnega signala

V zgornjem delu vmesnika prikazani časovni okvirji so oštevilčeni in se po višini prilagodijo glede na amplitudno ovojnico. Okvir lahko izberemo s poljubnim miškinim gumbom. Če to storimo z levim gumbom, potem izberemo en sam okvir. Njegov magnitudni spekter se prikaže v spodnjem delu okna. Z desnim ali srednjim miškinim gumbom lahko izberemo zaporedje okvirjev, in sicer izberemo zadnjega v zaporedju. Prvi v zaporedju je tisti, ki je bil predhodno izbran. Ob izbiri zaporedja okvirjev se predvaja zvočni signal, ki časovno ustreza izbranemu zaporedju okvirjev. Če je okvir časovno prekrak, se posnetek predvaja z razpolovljeno frekvenco vzorčenja, torej dvakrat počasneje in oktavo nižje. Med predvajanjem se po celotni višini področja valovne krivulje premika navpična rdeča črta, ki označuje mesto predvajanja.

Nad zgornjim delom področja časovnega signala se prikaže status izbranega okvirja: številka okvirja, oz. razpon izbranih okvirjev (vidno kot **Selected frame**). Če je izbran en sam okvir, se izpiše še število pravilno prepoznanih tonov (prikazano kot **ncors**), število referenčnih tonov (**nref**) ter število tonov, ki jih je zaznal algoritem (**nsys**).

Časovni okvirji so obarvani glede na pravilnost detekcije. To omogoča hitro iskanje delov posnetka, kjer detekcija deluje slabo. Povsem modri okvirji označujejo dobro detekcijo, povsem rdeči pa slabo. Sivi in rjavi so nekeje

vmes. Slika 5.1 prikazuje izsek signala. Opazimo, da so najbolj rdeče obarvani okvirji na koncih in začetkih kitic, največ modrih pa je na sredini.

5.1.2 Prikaz magnitudnega spektra

Spodnji del grafičnega vmesnika je namenjen prikazu magnitudnega spektra izbranega okvirja. Pod spektrogramom so označene frekvence, nad njim pa na ustreznih frekvenčnih mestih imena tonov brez oktavne informacije. Osrednje frekvence tonov so kot navpične rdeče črte prikazane tudi v ozadju spektrograma. Trikotniki označujejo sklop filtrov, ki so v Klapurijevem algoritmu namenjeni zajemanju energije spektra pri spektralnem beljenju.

V spektrogramu je poleg velikega modrega spektrograma v ospredju prikazan še manjši zelen, ki označuje prepoznan zvok. Pod spektrogramom se nahajata dva smerna gumba, s katerima lahko opazujemo potek delovanja algoritma iterativne detekcije in odštevanja. Klik na gumb v desno prikaže naslednji ostanek magnitudnega spektra in naslednji prepoznan zvok. Klik na gumb v levo deluje podobno, le v obratni smeri. Med obema gumboma se nahaja oznaka, ki šteje zaporedni prikazani ostanek. Kako zgleda celoten vmesnik, si je moč ogledati na sliki 5.2.

5.2 Analiza napak

Čeprav je grafični vmesnik zelo preprost, pa vseeno nudi osnoven vpogled v delovanje Klapurijevega algoritma. V tem razdelku si pogledamo primer, ko sistem razpozna napačne tone, napake pa so tudi v referenčni transkripciji.

Pri transkripcije glasbe, tako ročni kot avtomatizirani, so napake pogoste. Za to je več razlogov. Eden izmed njih je, da je notni ali drug simbolni zapis precej omejen, zato je določene glasbene prvine z njimi težko, če že ne nemogoče, izraziti. Drug razlog je nagnjenost k poenostavljanju, kar je v večini primerov bolj prednost kot slabost. Razloge za napake se da najti tudi v pomanjkljivem muzikološkem znanju ali pa preprosto zato, ker je transkripcija težek problem.

Do napake lahko pride tudi pri pretvorbi MIDI dogodkov v časovne okvirje, saj so v okvir vključeni vsi toni, ki vsaj malo časovno sovpadajo z okvirjem. Včasih so to kratki deli konca tona, ki zvenijo že od prej in iz-zvenijo v tem okvirju, spet drugič se lahko ta ton šele začne. Takšne tone, ki časovno ne prekrivajo zadosten del okvirja, bi lahko pri pretvorbi iz MIDI dogodkov v časovne okvirje odstranili. Algoritem za pretvorbo ta delež že izračuna, vendar ga zaenkrat ne upošteva.

V našem primeru so napake v referenci zlasti pogoste na začetkih in koncih kitic. Vzroki za napake na začetku kitic so zaradi nenatančne poravnave med MIDIjem in zvočnim posnetkom, zaradi nestabilnih in neprepričljivih vstopov pevcev, zaradi poenostavitve transkripcije in tudi prej omenjene pretvorbe MIDI dogodkov v časovne okvirje.

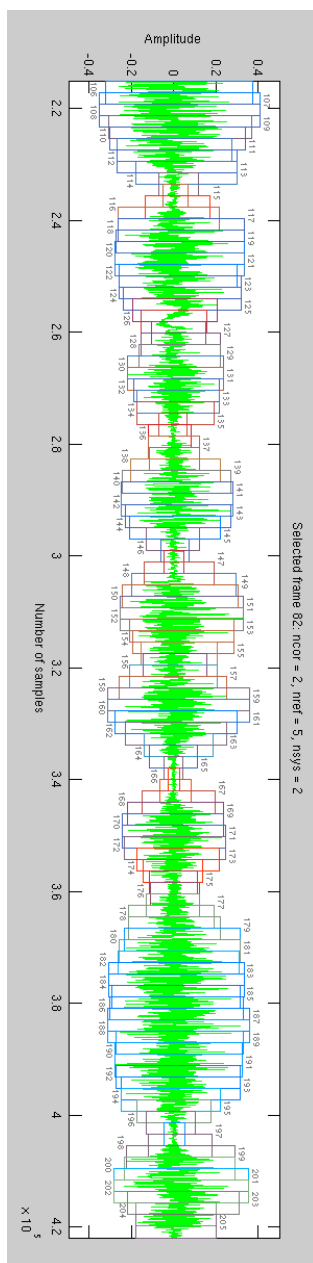
Poglejmo primer časovnega okvirja, ko je referenčna transkripcija napačna, algoritem pa vrne pravilno. Zgodi se ob pričetku kitice, ko v dejanskem posnetku še nista aktivna oba glasova, pač pa samo eden, intonacija edinega aktivnega glasu pa je nekje približno vmes med $g\#$ in a in v naslednjih časovnih okvirjih preide v a . MIDI za ta časovni okvir vsebuje dvoglasje a , $c1\#$. Algoritem najde najizrazitejši ton za ta okvir pri frekvenci 212 Hz, kar je bližje $g\#$ pri 207,65 Hz kot a z 220. V magnitudnem spektru (desni del slike 5.3) sta vidna modro obarvana vrhova tako pri $g\#$ kot pri a , vendar se zeleno obarvan vrh, ki označuje prvi delni harmonični ton prepoznanega zvoka, nahaja jasno pri $g\#$.

S klikom na gumb za prikaz naslednje iteracije (prikazan na sredini na sliki 5.3) je prepoznani ton še vedno $g\#$, vendar je tokrat prvi delni harmonični ton pomaknjen bližje k a , zato je zgolj na osnovi prvega delnega harmoničnega tona težko določiti pravega. Že drugi, tretji in četrti harmonični delni ton, ki so zaporedoma najbližje tonom $g1\#$, $d2\#$ in $g2\#$, potrdijo, da je $g\#$ primernejša izbira za $F0$ kot a . Ker se ton, zaznan v drugi iteraciji, od prvega razlikuje za manj kot četrt tona, ga algoritem ne vključi v izhodno množico tonov, vendar ga vseeno odšteje iz spektra.

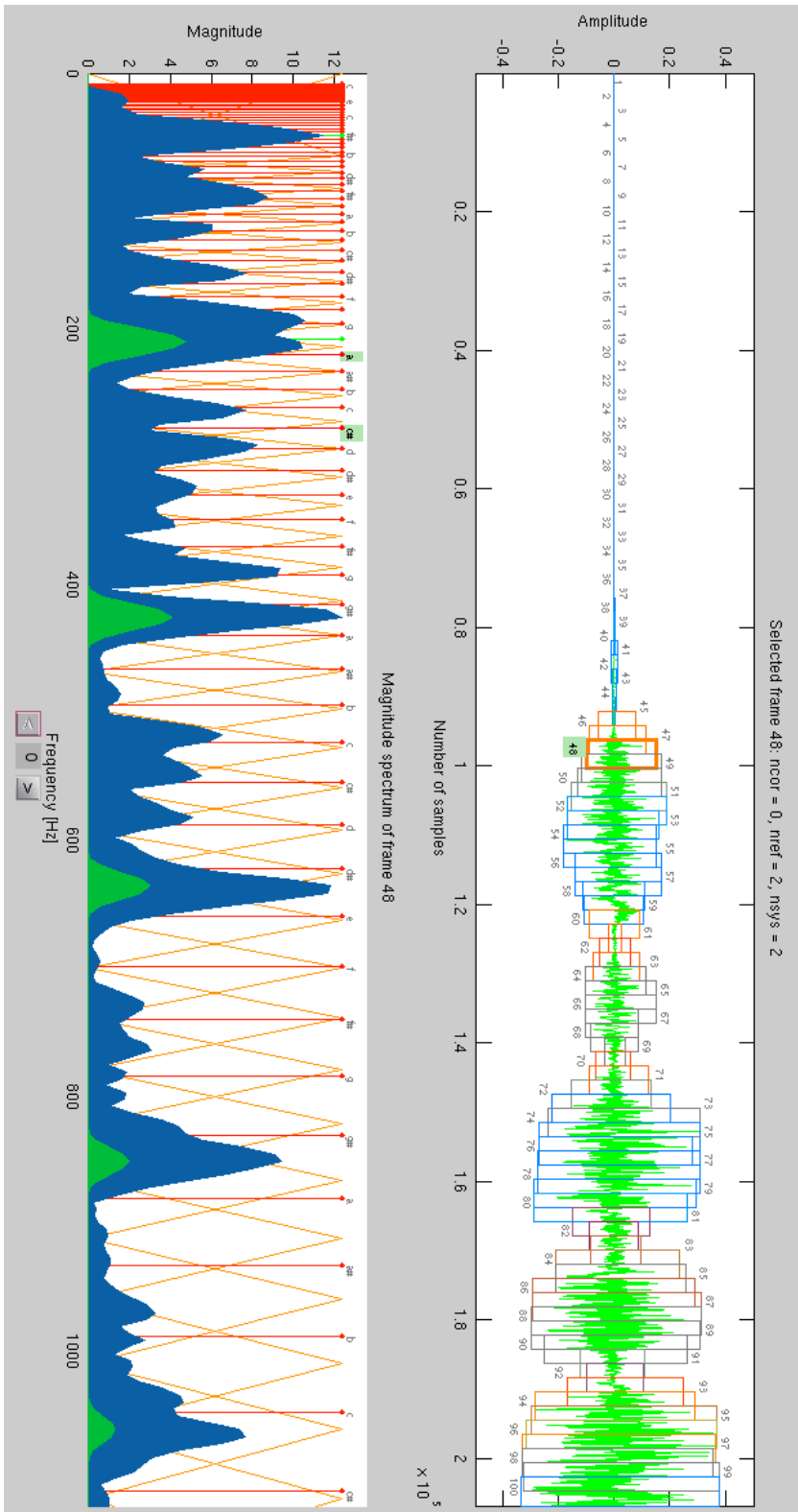
Z ogledom naslednje iteracije (levi del slike 5.3) se vidi, da tokrat algori-

tem zazna neobstoječ ton G1 s frekvenco 48,7 Hz, ki ima kar visoko energijo. Razlog za to lahko delno pripišemo učinku spektralnega beljenja, saj le-ta poudari vrhove pri nizkih frekvencah. Vseeno pa se delni harmonični toni zaznanega zvoka v glavnem ujemajo z modrimi vrhovi, zato beljenje ni edini razlog.

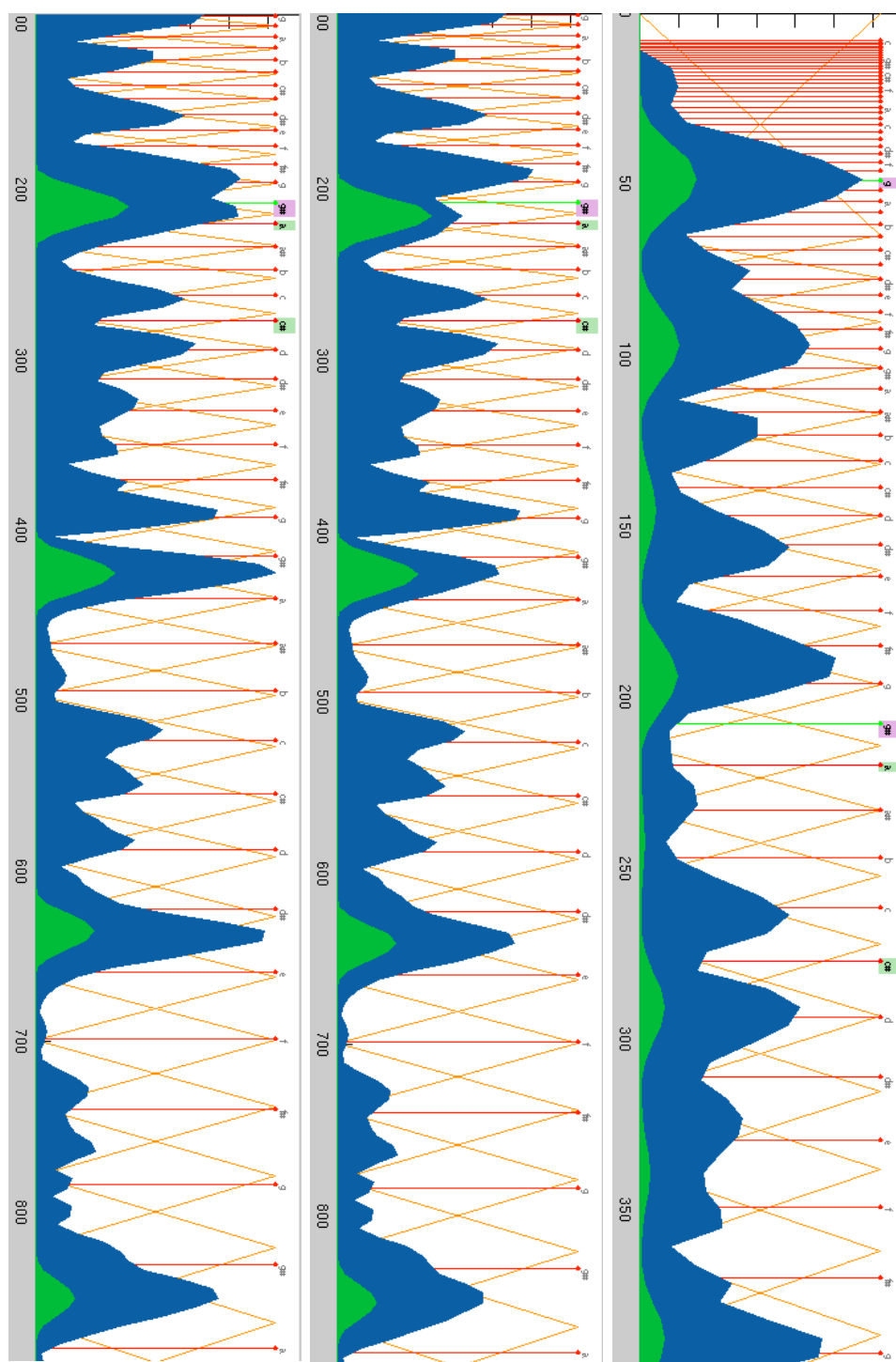
Iteracija se še vedno nadaljuje, vendar pa energija zaznanih tonov ni več visoka. Sicer pa lahko pri vseh zaznanih tonih opazimo, da so zeleni vrhovi v primerjavi z modrimi vsaj dvakrat nižji, čeprav v tem primeru, ko je v posnetku samo en glas, ni razloga, da bi algoritem odšteval tako zadržano.



Slika 5.1: Prikaz časovnega signala z obarvanimi okvirji.



Slika 5.2: Prikaz časovnega signala in magnitudnega spektra v izbranem okvirju.



Slika 5.3: Primer napačne referenčne transkripcije. Na desni in srednji sliki algoritem zazna ton $g\sharp$, ki je pravilen, v referenci pa je navedena terca a , $c1\sharp$. Na levi sliki algoritem zazna ton $G1$, ki ga ni niti v referenci niti dejansko v zvočnem posnetku.

Poglavje 6

Rezultati

V tem poglavju najprej opišemo, kako smo med sabo primerjali referenčne MIDI tone s toni, dobljenimi z algoritmom. Nato naštejemo uporabljene mere ocenjevanja. Nazadnje prikažemo rezultate algoritma in poskuse izboljšav.

6.1 Preurejanje podatkov pred vrednotenjem

V tem razdelku si pogledimo, kako pripravimo referenčne MIDI datoteke in izhodne frekvence Klapurijevega algoritma, da jih lahko med sabo primerjamo.

6.1.1 Pretvorba MIDI datoteke v časovne okvirje

Klapurijev algoritem vrne seznam časovnih okvirjev, v katerih so za pripadajoči čas prepoznane osnovne frekvence, skupaj z njihovo močjo (*salience*). Želimo, da so referenčni podatki v enaki obliki. Iz referenčnega MIDI zapisa s knjižnico MIDI Toolbox [5] in lastnega programa pridobimo matriko not. Posamezna vrstica v matriki not vsebuje sedem elementov. Ti so: čas v številu dob, trajanje v številu dob, številka kanala, višina tona, hitrost, čas pojavitve v sekundah in trajanje v sekundah. Izmed sedmerice so zanimivi trije: višina tona, čas pojavitve in trajanje.

Dva izmed vhodnih parametrov Klapurijevega algoritma sta velikost okna (K) in velikost skoka (K_{hop}), tj. premika tega okna, oba v številu vzorcev. Z deljenjem teh dveh vrednosti s frekvenco vzorčenja f_s ju pretvorimo v čas v sekundah. Od tod poznamo časovno lokalizacijo posameznih izhodnih okvirjev.

Za MIDI si pripravimo enake časovne okvirje in vanje na ustrezne časovne lokacije umestimo MIDI tone. Če časovni okvir n nastopi v času t_n^\perp in se konča v času t_n^\top , potem v časovni okvir n spadajo vsi MIDI toni z začetkom pred t_n^\top in koncem kasnejšim od t_n^\perp . Ko to storimo za vse okvirje, je MIDI strukturno enak obliki izhoda iz Klapurijevega algoritma. V nadaljevanju pretvorimo še vsebino.

6.1.2 Primerjava med frekvenco in MIDI številom

Elementi izhoda iz Klapurijevega algoritma so frekvence v nekem podanem razponu iskanja F0, medtem ko MIDI vsebuje MIDI tone, ki so vrednosti od 0 do 127. Torej je potrebno pretvoriti še frekvence v MIDI tone. Tukaj vpeljemo nekaj odstopanja. Dovolimo, da F0 od tona, ki ga določa MIDI število, odstopa za največ četrto tona. MIDI višina 69 ustreza komornemu tonu a1 pri 440 Hz. Frekvence tonov pri temperirani uglasitvi izračunamo po formuli

$$f = 440 \cdot 2^{\frac{p}{12}}, \quad (6.1)$$

kjer je p število poltonov stran od tona a1. Za pretvorbo frekvence v število poltonov od neke bazne frekvence uporabimo inverz od (6.1):

$$p = 12 \log_2 \frac{f}{440}.$$

Frekvenca f se ujema z MIDI tonom p , če velja:

$$|p - 69 - 12 \log_2 \frac{f}{440}| < 0,5.$$

V enotah cent to pomeni ujemanje na ± 50 centov. Cent je definiran tako, da je frekvenčna razdalja med poljubnima dvema frekvencama s poltonom razlike enaka 100.

6.2 Mere vrednotenja

Vrednotenje poteka neodvisno po posameznih istoležnih časovnih okvirjih. Naj $TP = N_{\text{cor}}$ označuje število vseh pravilno prepoznanih tonov, $N_{\text{sys}} = TP + FP$ število vseh tonov sistema, ki ga testiramo (Klapurijev algoritem) in $N_{\text{ref}} = TP + FN$ število vseh tonov v referenčni MIDI datoteki. Od tod definiramo naslednje mere:

$$\begin{aligned} \text{Pr} &= \frac{N_{\text{cor}}}{N_{\text{sys}}} = \frac{TP}{TP + FP}, \\ \text{R} &= \frac{N_{\text{cor}}}{N_{\text{ref}}} = \frac{TP}{TP + FN}, \\ \text{A} &= \frac{N_{\text{cor}}}{N_{\text{sys}} + N_{\text{ref}} - N_{\text{cor}}} = \frac{TP}{TP + FP + FN}, \\ \text{F} &= 2 \frac{\text{Pr} \cdot \text{R}}{\text{Pr} + \text{R}}, \\ \text{E}_{\text{tot}} &= \frac{\sum (\max\{N_{\text{ref}}, N_{\text{sys}}\} - N_{\text{cor}})}{\sum N_{\text{ref}}}, \\ \text{E}_{\text{sub}} &= \frac{\sum (\min\{N_{\text{ref}}, N_{\text{sys}}\} - N_{\text{cor}})}{\sum N_{\text{ref}}}, \\ \text{E}_{\text{mis}} &= \frac{\sum (\max\{0, N_{\text{ref}} - N_{\text{sys}}\})}{\sum N_{\text{ref}}}, \\ \text{E}_{\text{fa}} &= \frac{\sum (\max\{0, N_{\text{sys}} - N_{\text{ref}}\})}{\sum N_{\text{ref}}}. \end{aligned}$$

Mera Pr je delež pravilno prepoznanih tonov v sistemu, ne glede na to, koliko tonov je izpuščenih. Mera R je delež pravilnih detekcij glede na vse tone v referenci. Pri tem število odvečnih (nepravilnih) detekcij ne šteje. Mera A upošteva napačne detekcije obeh vrst, prav tako jih F ocena, ki je kvocient

med kvadratom geometrijske in aritmetične sredine.

Preostale štiri mere, $E_{(\cdot)}$, so definirane v [15]. E_{tot} je mera skupne napake, E_{sub} šteje, v razmerju glede na število vseh tonov v referenci, koliko zamenjav z ustreznim tonom bi lahko uporabili, preden bi bodisi v sistemu bodisi v referenci zmanjkalo tonov. E_{mis} je delež premajhnega števila detekcij glede na referenco. Nasprotno je E_{fa} delež prevelikega števila detekcij glede na referenco.

Mere Pr, R, A in F želimo čim večje, mere za napako pa čim manjše.

6.3 Evalvacija

V tem razdelku predstavimo rezultate delovanja Klapurijevega algoritma na množici ljudskih pesmi. Najprej pa opišemo parametre algoritma in njihove uporabljene vrednosti.

6.3.1 Parametri Klapurijevega algoritma

Klapurijev algoritem z izbiro parametrov omogoča določene prilagoditve glede na izbrano testno množico. Večino teh parametrov je izpostavil že originalen avtorjev algoritem, določeni pa so bili izpostavljeni naknadno.

Poleg zvočnega signala s frekvenco vzorčenja $f_s = 44,1$ kHz algoritem sprejme še parametre, prikazane v tabeli 6.1. Parameter K označuje velikost časovnega okna v številu vzorcev. Vrednost 2^{12} vzorcev pri frekvenci vzorčenja f_s se pretvori v približno 93 ms dolgo okno. H pomeni, koliko vzorcev algoritem preskoči do naslednjega okna. P označuje največjo polifonijo. Parametri τ_{min} , τ_{max} in τ_{prec} predpisujejo frekvenčno območje in natančnost iskanja F0. Privzete vrednosti parametrov iskanje F0 omejujejo na območje 40 – 2205 Hz. Takšen razpon je za vokalne glasove, še zlasti ljudsko petje, nepotreben. Območje iskanja smo zmanjšali na 80 – 1102 Hz, kar je rahlo izboljšalo rezultat, saj je prišlo do kar nekaj napak v območju med 40 in 80 Hz. Zadnje omenjeno območje se ujema z vrednostmi za dolžino periode v tabeli. Parametri α , γ in ρ upravljajo utežno funkcijo w . Uporabljene so

bile kar privzete vrednosti. Vedeti moramo, da sta bili utežitveni funkciji w_1 in w_2 , ki sestavljata w , s strani avtorja optimizirani na večjem številu testnih primerov, spreminjanje posameznih parametrov pa ni prineslo izboljšav. Parameter ν označuje količino spektralnega beljenja. Praviloma vrednost 0,1 na naši testni množici daje nekoliko boljše rezultate kot privzeta vrednost 0,33. Nekoliko več beljenja ima torej praviloma pozitiven učinek. Parameter win določa uporabljeno okensko funkcijo pri računanju FFT kot tudi pri gradnji prepoznanih spektralnih komponent, ki se jih odšteva iz magnitudnega spektra. Zamenjava okenske funkcije ni pokazala kakšnih opaznih razlik. V nadaljevanju se bomo na to izbiro parametrov sklicevali kot na privzete parametre.

Parameter	Vrednost
K	2^{12}
K_{hop}	2^{11}
P	8
τ_{min}	40
τ_{max}	551
τ_{prec}	0,1
α	52
γ	0,1
ρ	1,4
ν	0,1
win	Hann

Tabela 6.1: Izbrane vrednosti parametrov Klapurijevega algoritma.

6.3.2 Določanje polifonije

Klapurijev algoritem za vsak časovni okvir vrne P tonov skupaj z njihovimi izračunanimi energijami, četudi se v trenutnem časovnem okvirju nahaja tišina. Določanje polifonije se izkaže kot težek problem. Predlagana Klapurijeva rešitev iz razdelka 3.5.5 se ni izkazala preveč dobro. Preprosto upravljanje se je obneslo bolje. Izkustveno je bilo ugotovljeno, da se vrednosti izhodnih energij, gledane skozi vse okvirje, približno prilegajo logit

funkciji. Optimalna vrednost upragovljenja se praviloma nahaja na tistem delu logit funkcije, kjer začne po položnem delu zopet hitro naraščati. Mera za določanje optimalne vrednosti je bil produkt mer Pr , R in A .

6.3.3 Uspešnost algoritma

Klapurijev algoritem smo preizkusili na testni množici slovenskih ljudskih pesmi. Uporabljeni so bili parametri iz tabele 6.1. Izhodne frekvence smo zaokrožili k najbližjemu tonu. Opravili smo tri vrste vrednotenja. Prvo vrednotenje neposredno primerja referenčne tone z izhodnimi. Drugo vrednotenje zanemari oktavno informacijo. Tretje vrednotenje šteje kot pravilno prepoznani vsak ton, ki je največ poltona stran od kateregakoli referenčnega tona.

Prvo vrednotenje

Rezultati v tabeli 6.2 niso preveč dobri. Nizke vrednosti napak E_{fa} in večje E_{mis} kažejo na to, da je fiksno upragovljenje nastavljeno nekoliko bolj agresivno. Prepoznana stopnja polifonije v časovnih okvirjih je zato v glavnem nižja od referenčne. Z nižanjem praga bi znižali napako E_{mis} , vendar bi zaradi napačno prepoznanih na novo vključenih tonov povečali napako E_{sub} . Vsekakor tolikšna razlika med E_{mis} in E_{fa} kaže na neoptimalno detekcijo stopnje polifonije. Toda prag je bil optimiziran glede na produkt mer Pr , R in A , kar pomeni, da četudi bi bila razlika med omenjenima merama napake manjša, bi bil rezultat glede na produkt mer Pr , R in A slabši, torej bi se napaka E_{sub} povišala.

Algoritem se je najbolje izkazal pri posnetkih 2 in 21, najslabše pa pri 6., 7., 20., 24., 26., 29. in 31. posnetku. Kaj je posebnega pri 6.? Čeprav je dvoglasje in pojeta le ena ženska in moški, so rezultati sodeč po ocenah slabi. Problem je intonacija. Pri ročni transkripciji se je bilo potrebno odločiti za eno izmed dveh možnih intonacij, saj pevca ves čas pojeta nekje vmes med obema možnima intonacijama. Pričakovano je, da bi mehkejše vrednotenje,

ki dopušča do 100 centov napake namesto 50, moralo pokazati boljše rezultate. Če je to res, je bila pri ročni transkripciji tonaliteta slabo izbrana.

V 7. posnetku pojejo trije glasovi. Melodijo in terco nad njo pojeta dve ženski, moški poje bas. Gibanje basa obsega samo dva tona, toniko in dominantno. Izkaže se, da je tudi v tem primeru vzrok težavna določitev ustrezne intonacije, prav tako pri vseh ostalih s slabo oceno.

Drugo vrednotenje

Kot prikazuje tabela 6.3, zanemarjanje oktavne informacije pričakovano izboljša rezultate, vendar malo. Kot kaže, je $\hat{s}(\tau)$ dokaj robustna cenilka.

Tretje vrednotenje

Zaradi slabe intonacije v posnetkih se je to mehkejše vrednotenje izkazalo kot veliko izboljšanje, kot je razvidno iz tabele 6.4. Kot kaže je ogromno napak, ki so pol tona stran od prave vrednosti. Vendar pa je kljub obetavnim številkam potrebno opozoriti na naslednje. Denimo, da sta v nekem časovnem okvirju v referenci tona d in f, ki tvorita interval male terce. Vemo, da so terce v ljudskem petju pogoste. Kateri toni bodo glede na to mehkejše vrednotenje šteti kot pravilni? To bodo toni c#, d, d#, e, f in f#. Če algoritem najde denimo tona c# in f#, bo to štelo kot pravilno. Njuna intervalna razdalja je čista kvarta, kar je precej narobe. Bolj realno vrednotenje bi bilo takšno, ki bi poskusilo ohranjati intervalna razmerja med prepoznanimi toni. Če bi sistem najprej našel f#, bi to štelo kot pravilno, vendar naslednji ton bi bil pravilen le, če bi bil malo terco pod f#, torej ton d#, medtem ko d ali c# ne bi bila pravilna.

6.4 Poskusi izboljšave

Poskušanje različnih parametrov algoritma je časovno zamudno opravilo. Karkoli smo uspeli preizkusiti, so bili rezultati kvečjemu slabši. Vselej pa

se je spektralno beljenje izkazalo kot pomembno k prispevanju boljšega rezultata, zato smo izboljšave poskusili pridobiti na tem delu algoritma. Druge možnosti izboljšav so v utežitveni funkciji in je poleg beljenja prav tako predmet obravnave tega razdelka.

6.4.1 Spremembe spektralnega beljenja

Klapuri je kot privzet parameter spektralnega beljenja uporabil $\nu = 0,33$ in je bil fiksno vdelan v program. Spremenili smo ga v parameter in preizkusili nekaj vrednosti. Za dano podatkovno množico smo našli nekoliko boljšo vrednost. V večini primerov daje parameter 0,1 rahlo boljše rezultate, to pa pomeni več beljenja v spektru. Dodatno povečevanje količine beljenja pa je vodilo v negativen učinek. Če hočemo to razumeti, moramo boljše poznati učinke beljenja.

Učinek beljenja

Spomnimo, da beljenje uporabi nabor filtrov v frekvenčni domeni. Vsak filter ima obliko trikotnika. Po dualnosti med konvolucijo in Fourierjevo transformacijo je konvoluiranje v časovni domeni enakovredno množenju v frekvenčni, in obratno. Trikotniki pri nizkih frekvencah so ozki in zato iz spektra zajamejo malo energije. V nasprotju so filtri pri visokih frekvencah široki in zato pokrivajo večji del spektra. Toda s parametrom beljenja na intervalu $[0, 1)$, ki nastopi kot eksponent v potenciranju zajete energije, je učinek beljenja obraten količini zajete energije iz spektra, kar pomeni, da nizko energijo dvigne, visoko pa zaduši. To lahko ima do neke mere pozitiven učinek pri vokalni glasbi, zlasti zaradi vokalnih oblikovalcev. Spektralno beljenje nekoliko zmanjša učinek le-teh, zato delni harmonični toni okrog frekvenc, ki so pri določenem vokalu sicer poudarjeni, postanejo manj izraziti in se lepše prilegajo obstoječi utežitveni funkciji.

Ker povečevanje spektralnega beljenja deluje samo do neke mere, smo namesto povečevanja parametra beljenja poskusili beljenje izvesti dvakrat

zapored. To se je izkazalo kot rahla izboljšava, vendar vseeno premajhna za optimizem. Rezultati so prikazani v tabeli 6.5.

Spremembe utežitvene funkcije

Izboljšanje rezultatov smo poskusili doseči tudi s spremembo utežitvene funkcije. Recipročno linearno funkcijo frekvence w_2 smo spremenili v linearno funkcijo, ki smo jo definirali skozi točki pri ničelni in končni frekvenci prvotne w_2 . Takšna utežitvena funkcija deluje kot protiutež funkciji w_1 , kar povzroči konstanten odziv v magnitudnem spektru. Utežitev spektra je potem v celoti odvisna od spektralnega beljenja. Takšna sprememba je vodila do slabših rezultatov. Prikazani so v tabeli 6.6.

#	Pr	R	A	F	E _{tot}	E _{sub}	E _{mis}	E _{fa}
2	0,6891	0,5639	0,4495	0,6203	0,4895	0,2010	0,2351	0,0535
3	0,6147	0,4717	0,3640	0,5338	0,5537	0,2703	0,2580	0,0254
4	0,5496	0,4690	0,3388	0,5061	0,5877	0,3277	0,2033	0,0567
5	0,4996	0,4178	0,2946	0,4551	0,6446	0,3560	0,2262	0,0624
6	0,3515	0,2885	0,1883	0,3169	0,7655	0,4784	0,2331	0,0540
7	0,3942	0,3211	0,2150	0,3539	0,7428	0,4296	0,2492	0,0639
8	0,5402	0,4487	0,3247	0,4902	0,6333	0,2999	0,2514	0,0820
9	0,5951	0,5292	0,3891	0,5602	0,5595	0,2714	0,1994	0,0887
10	0,5283	0,4698	0,3309	0,4973	0,6288	0,3209	0,2093	0,0986
11	0,6595	0,5480	0,4272	0,5986	0,5235	0,2115	0,2405	0,0715
12	0,5390	0,4994	0,3499	0,5184	0,5983	0,3294	0,1712	0,0977
13	0,5755	0,5101	0,3707	0,5409	0,5709	0,2952	0,1947	0,0811
14	0,6644	0,5091	0,4050	0,5765	0,5155	0,2325	0,2584	0,0246
15	0,4957	0,3452	0,2555	0,4070	0,6977	0,3083	0,3465	0,0429
16	0,5998	0,4734	0,3597	0,5291	0,5861	0,2564	0,2702	0,0595
17	0,6005	0,4457	0,3438	0,5117	0,5863	0,2645	0,2898	0,0320
18	0,7002	0,5421	0,4400	0,6111	0,4831	0,2069	0,2510	0,0252
19	0,5319	0,4417	0,3181	0,4826	0,6243	0,3227	0,2355	0,0660
20	0,4568	0,3886	0,2658	0,4200	0,6950	0,3784	0,2330	0,0836
21	0,6839	0,5712	0,4519	0,6225	0,4519	0,2409	0,1879	0,0231
22	0,5178	0,4822	0,3328	0,4993	0,6118	0,3551	0,1627	0,0940
23	0,5975	0,4828	0,3643	0,5340	0,5406	0,3018	0,2154	0,0234
24	0,5472	0,2846	0,2304	0,3745	0,7468	0,2041	0,5113	0,0315
25	0,4883	0,4601	0,3105	0,4738	0,6414	0,3806	0,1593	0,1016
26	0,4005	0,4163	0,2565	0,4083	0,7585	0,4483	0,1354	0,1748
27	0,4675	0,4114	0,2801	0,4376	0,6776	0,3797	0,2089	0,0890
28	0,5134	0,4088	0,2947	0,4552	0,6451	0,3335	0,2576	0,0540
29	0,4032	0,4456	0,2685	0,4234	0,7931	0,4208	0,1336	0,2387
30	0,5967	0,5125	0,3806	0,5514	0,5519	0,2820	0,2055	0,0645
31	0,3848	0,3525	0,2255	0,3680	0,7668	0,4442	0,2032	0,1193
32	0,6242	0,5461	0,4110	0,5825	0,5472	0,2355	0,2184	0,0933
33	0,5785	0,4822	0,3568	0,5260	0,5833	0,2859	0,2320	0,0654
34	0,5069	0,5306	0,3500	0,5185	0,6603	0,3252	0,1441	0,1909
35	0,5354	0,4959	0,3467	0,5149	0,6306	0,3038	0,2003	0,1265
36	0,5732	0,4817	0,3545	0,5235	0,5971	0,2799	0,2384	0,0789
37	0,6163	0,5154	0,3902	0,5613	0,5292	0,2763	0,2083	0,0446
38	0,5862	0,4963	0,3676	0,5375	0,5732	0,2808	0,2228	0,0695
\bar{x}	0,5461	0,4611	0,3352	0,4984	0,6160	0,3119	0,2271	0,0771

Tabela 6.2: Uspešnost Klapurijevega algoritma pri privzetih parametrih.

#	Pr	R	A	F	E _{tot}	E _{sub}	E _{mis}	E _{fa}
2	0,7298	0,6339	0,5134	0,6785	0,4275	0,1733	0,2351	0,0535
3	0,6423	0,5451	0,4181	0,5897	0,5119	0,2466	0,2580	0,0254
4	0,5784	0,5160	0,3749	0,5454	0,5604	0,2998	0,2033	0,0567
5	0,5440	0,4939	0,3493	0,5178	0,6116	0,3086	0,2262	0,0624
6	0,3688	0,3361	0,2134	0,3517	0,7591	0,4801	0,2331	0,0540
7	0,4325	0,3634	0,2461	0,3949	0,7074	0,4060	0,2492	0,0639
8	0,5941	0,5007	0,3731	0,5434	0,5793	0,2621	0,2514	0,0820
9	0,6316	0,5452	0,4137	0,5852	0,5206	0,2522	0,1994	0,0887
10	0,5614	0,4915	0,3551	0,5241	0,5810	0,3116	0,2093	0,0986
11	0,6924	0,5594	0,4480	0,6188	0,4901	0,1991	0,2405	0,0715
12	0,5593	0,5062	0,3619	0,5315	0,5730	0,3195	0,1712	0,0977
13	0,5961	0,5190	0,3840	0,5549	0,5468	0,2858	0,1947	0,0811
14	0,6952	0,5179	0,4221	0,5936	0,4938	0,2154	0,2584	0,0246
15	0,5173	0,3530	0,2655	0,4196	0,6803	0,2961	0,3465	0,0429
16	0,6366	0,4850	0,3798	0,5505	0,5489	0,2430	0,2702	0,0595
17	0,6241	0,4559	0,3577	0,5269	0,5597	0,2590	0,2898	0,0320
18	0,7258	0,5513	0,4562	0,6266	0,4610	0,1960	0,2510	0,0252
19	0,5287	0,5282	0,3591	0,5284	0,6329	0,3098	0,2355	0,0660
20	0,4852	0,4612	0,3097	0,4729	0,6734	0,3547	0,2330	0,0836
21	0,6968	0,5729	0,4586	0,6288	0,4409	0,2354	0,1879	0,0231
22	0,5607	0,5301	0,3745	0,5450	0,5615	0,3237	0,1627	0,0940
23	0,6125	0,4858	0,3716	0,5419	0,5231	0,2984	0,2154	0,0234
24	0,5896	0,3030	0,2502	0,4003	0,7237	0,1843	0,5113	0,0315
25	0,5212	0,4741	0,3303	0,4965	0,5980	0,3634	0,1593	0,1016
26	0,4727	0,5047	0,3229	0,4882	0,6859	0,3723	0,1354	0,1748
27	0,5060	0,5547	0,3599	0,5293	0,6637	0,3231	0,2089	0,0890
28	0,5747	0,4550	0,3404	0,5079	0,5847	0,2970	0,2576	0,0540
29	0,4779	0,5580	0,3467	0,5149	0,7280	0,3236	0,1336	0,2387
30	0,6409	0,6008	0,4495	0,6202	0,4849	0,2509	0,2055	0,0645
31	0,4342	0,4232	0,2728	0,4286	0,7233	0,4049	0,2032	0,1193
32	0,6866	0,5719	0,4535	0,6240	0,4864	0,2029	0,2184	0,0933
33	0,6123	0,4915	0,3748	0,5453	0,5473	0,2724	0,2320	0,0654
34	0,5665	0,5631	0,3935	0,5648	0,5812	0,2866	0,1441	0,1909
35	0,5925	0,5167	0,3812	0,5520	0,5642	0,2744	0,2003	0,1265
36	0,6111	0,4938	0,3757	0,5462	0,5562	0,2643	0,2384	0,0789
37	0,6454	0,5218	0,4055	0,5771	0,4969	0,2681	0,2083	0,0446
38	0,6285	0,6049	0,4456	0,6165	0,5144	0,2383	0,2228	0,0695
\bar{x}	0,5831	0,5024	0,3705	0,5373	0,5779	0,2866	0,2271	0,0771

Tabela 6.3: Uspešnost Klapurijevega algoritma pri privzetih parametrih, kadar zanemarimo oktavno informacijo.

#	Pr	R	A	F	E _{tot}	E _{sub}	E _{mis}	E _{fa}
2	0,8792	0,7195	0,6548	0,7914	0,3340	0,0454	0,2351	0,0535
3	0,8886	0,6819	0,6282	0,7716	0,3435	0,0601	0,2580	0,0254
4	0,8623	0,7358	0,6584	0,7940	0,3208	0,0609	0,2033	0,0567
5	0,7592	0,6349	0,5285	0,6915	0,4276	0,1389	0,2262	0,0624
6	0,7677	0,6302	0,5293	0,6922	0,4238	0,1367	0,2331	0,0540
7	0,7549	0,6150	0,5127	0,6778	0,4489	0,1357	0,2492	0,0639
8	0,7827	0,6501	0,5507	0,7103	0,4319	0,0985	0,2514	0,0820
9	0,8228	0,7318	0,6322	0,7746	0,3569	0,0688	0,1994	0,0887
10	0,8277	0,7360	0,6382	0,7792	0,3626	0,0547	0,2093	0,0986
11	0,8688	0,7220	0,6510	0,7886	0,3496	0,0375	0,2405	0,0715
12	0,8364	0,7749	0,6730	0,8045	0,3227	0,0539	0,1712	0,0977
13	0,8494	0,7529	0,6642	0,7982	0,3282	0,0524	0,1947	0,0811
14	0,9077	0,6955	0,6495	0,7875	0,3291	0,0461	0,2584	0,0246
15	0,8277	0,5764	0,5146	0,6796	0,4665	0,0771	0,3465	0,0429
16	0,8619	0,6803	0,6135	0,7604	0,3792	0,0495	0,2702	0,0595
17	0,8932	0,6630	0,6143	0,7611	0,3690	0,0472	0,2898	0,0320
18	0,9300	0,7200	0,6830	0,8117	0,3052	0,0290	0,2510	0,0252
19	0,8389	0,6968	0,6146	0,7613	0,3693	0,0677	0,2355	0,0660
20	0,7452	0,6339	0,5210	0,6851	0,4497	0,1331	0,2330	0,0836
21	0,9245	0,7721	0,7263	0,8414	0,2510	0,0400	0,1879	0,0231
22	0,8347	0,7774	0,6736	0,8050	0,3167	0,0599	0,1627	0,0940
23	0,9009	0,7279	0,6740	0,8052	0,2955	0,0567	0,2154	0,0234
24	0,8439	0,4390	0,4060	0,5776	0,5925	0,0497	0,5113	0,0315
25	0,8012	0,7550	0,6358	0,7774	0,3466	0,0858	0,1593	0,1016
26	0,6872	0,7143	0,5390	0,7005	0,4605	0,1503	0,1354	0,1748
27	0,7518	0,6617	0,5431	0,7039	0,4273	0,1294	0,2089	0,0890
28	0,8117	0,6464	0,5621	0,7197	0,4076	0,0959	0,2576	0,0540
29	0,6413	0,7088	0,5076	0,6734	0,5299	0,1577	0,1336	0,2387
30	0,8213	0,7054	0,6116	0,7590	0,3590	0,0890	0,2055	0,0645
31	0,7325	0,6710	0,5389	0,7004	0,4483	0,1258	0,2032	0,1193
32	0,8290	0,7252	0,6308	0,7736	0,3681	0,0564	0,2184	0,0933
33	0,8603	0,7170	0,6422	0,7822	0,3484	0,0510	0,2320	0,0654
34	0,7557	0,7910	0,6299	0,7730	0,3999	0,0648	0,1441	0,1909
35	0,7852	0,7272	0,6066	0,7551	0,3993	0,0724	0,2003	0,1265
36	0,8446	0,7099	0,6279	0,7714	0,3690	0,0517	0,2384	0,0789
37	0,8888	0,7433	0,6801	0,8096	0,3013	0,0483	0,2083	0,0446
38	0,8051	0,6817	0,5851	0,7383	0,3879	0,0955	0,2228	0,0695
\bar{x}	0,8223	0,6953	0,6041	0,7510	0,3818	0,0777	0,2271	0,0771

Tabela 6.4: Uspešnost Klapurijevega algoritma pri privzetih parametrih, kadar kot napačne štejemo samo tone, ki so od tona v referenci oddaljeni več kot pol tona.

#	Pr	R	A	F	E _{tot}	E _{sub}	E _{mis}	E _{fa}
2	0,6723	0,5801	0,4522	0,6228	0,4888	0,2139	0,2061	0,0689
3	0,6111	0,4749	0,3647	0,5345	0,5535	0,2737	0,2514	0,0284
4	0,5658	0,4635	0,3419	0,5096	0,5778	0,3144	0,2221	0,0413
5	0,5169	0,4064	0,2946	0,4551	0,6390	0,3343	0,2592	0,0455
6	0,3735	0,2863	0,1934	0,3241	0,7522	0,4418	0,2720	0,0385
7	0,4161	0,3018	0,2120	0,3498	0,7321	0,3895	0,3087	0,0339
8	0,5537	0,4484	0,3294	0,4955	0,6232	0,2897	0,2619	0,0716
9	0,5930	0,5604	0,4047	0,5762	0,5320	0,2922	0,1475	0,0924
10	0,5576	0,4686	0,3416	0,5093	0,5716	0,3315	0,1998	0,0402
11	0,6584	0,5608	0,4344	0,6057	0,4738	0,2563	0,1829	0,0346
12	0,5719	0,4747	0,3503	0,5188	0,5715	0,3091	0,2161	0,0462
13	0,5637	0,5300	0,3758	0,5463	0,5837	0,2965	0,1736	0,1137
14	0,6688	0,5222	0,4150	0,5865	0,5033	0,2330	0,2447	0,0255
15	0,4981	0,3689	0,2689	0,4239	0,6683	0,3345	0,2966	0,0372
16	0,5814	0,4898	0,3621	0,5317	0,5733	0,2895	0,2207	0,0632
17	0,6107	0,4357	0,3410	0,5086	0,5893	0,2528	0,3115	0,0250
18	0,6993	0,5512	0,4456	0,6165	0,4823	0,2036	0,2452	0,0335
19	0,5623	0,4297	0,3220	0,4871	0,6043	0,3005	0,2698	0,0340
20	0,4424	0,4025	0,2670	0,4215	0,7079	0,3969	0,2006	0,1104
21	0,6768	0,5814	0,4551	0,6255	0,4331	0,2631	0,1555	0,0145
22	0,5272	0,4766	0,3339	0,5007	0,6008	0,3500	0,1734	0,0775
23	0,5974	0,4907	0,3688	0,5388	0,5280	0,3120	0,1973	0,0188
24	0,5610	0,2798	0,2296	0,3734	0,7438	0,1953	0,5248	0,0236
25	0,5021	0,4436	0,3081	0,4710	0,6275	0,3688	0,1876	0,0710
26	0,4096	0,4184	0,2610	0,4140	0,7548	0,4298	0,1517	0,1732
27	0,4660	0,4068	0,2775	0,4344	0,6816	0,3778	0,2154	0,0884
28	0,5207	0,4174	0,3016	0,4634	0,6383	0,3285	0,2540	0,0557
29	0,4262	0,4335	0,2737	0,4298	0,7502	0,4000	0,1665	0,1837
30	0,5922	0,5203	0,3830	0,5539	0,5522	0,2857	0,1940	0,0725
31	0,3954	0,3509	0,2283	0,3718	0,7531	0,4326	0,2165	0,1040
32	0,6357	0,5608	0,4244	0,5959	0,4902	0,2704	0,1688	0,0511
33	0,5527	0,5031	0,3575	0,5267	0,5532	0,3508	0,1461	0,0562
34	0,5555	0,5062	0,3603	0,5297	0,5474	0,3514	0,1423	0,0536
35	0,5369	0,5167	0,3574	0,5266	0,5916	0,3373	0,1460	0,1083
36	0,5805	0,4840	0,3586	0,5279	0,5613	0,3044	0,2116	0,0453
37	0,6145	0,5158	0,3897	0,5608	0,5293	0,2785	0,2057	0,0451
38	0,5497	0,5337	0,3714	0,5416	0,5965	0,3070	0,1593	0,1302
\bar{x}	0,5518	0,4648	0,3394	0,5030	0,5989	0,3161	0,2191	0,0637

Tabela 6.5: Uspešnost Klapurijevega algoritma pri privzetih parametrih, kadar spektralno beljenje izvedemo dvakrat.

#	Pr	R	A	F	E _{tot}	E _{sub}	E _{mis}	E _{fa}
2	0,5641	0,4486	0,3331	0,4998	0,6073	0,2907	0,2607	0,0559
3	0,5517	0,3006	0,2416	0,3892	0,7032	0,2404	0,4590	0,0038
4	0,4309	0,3255	0,2276	0,3708	0,7137	0,3907	0,2838	0,0392
5	0,5333	0,3114	0,2447	0,3932	0,6999	0,2612	0,4274	0,0113
6	0,3478	0,2586	0,1741	0,2966	0,7963	0,4300	0,3114	0,0548
7	0,3173	0,2912	0,1790	0,3037	0,8864	0,4489	0,2599	0,1777
8	0,3280	0,3027	0,1868	0,3149	0,8537	0,4638	0,2335	0,1564
9	0,5594	0,5063	0,3620	0,5315	0,5699	0,3225	0,1712	0,0762
10	0,5350	0,3234	0,2525	0,4031	0,6897	0,2680	0,4086	0,0132
11	0,5894	0,5161	0,3796	0,5503	0,5363	0,3072	0,1767	0,0524
12	0,4857	0,4714	0,3144	0,4784	0,6267	0,4011	0,1276	0,0980
13	0,5300	0,4651	0,3293	0,4955	0,5950	0,3523	0,1826	0,0602
14	0,6053	0,4603	0,3540	0,5229	0,5763	0,2635	0,2762	0,0366
15	0,4578	0,3727	0,2586	0,4109	0,6839	0,3848	0,2425	0,0566
16	0,5411	0,4957	0,3490	0,5174	0,5805	0,3443	0,1600	0,0762
17	0,5728	0,4667	0,3462	0,5143	0,5526	0,3288	0,2045	0,0192
18	0,6517	0,5555	0,4283	0,5998	0,4994	0,2420	0,2025	0,0549
19	0,3983	0,3776	0,2404	0,3876	0,8001	0,3928	0,2296	0,1777
20	0,3796	0,2755	0,1900	0,3193	0,7616	0,4130	0,3115	0,0371
21	0,6634	0,5924	0,4555	0,6259	0,4317	0,2766	0,1311	0,0240
22	0,4315	0,3760	0,2514	0,4018	0,7004	0,4190	0,2050	0,0764
23	0,5706	0,4750	0,3499	0,5184	0,5375	0,3449	0,1801	0,0125
24	0,4606	0,2508	0,1938	0,3247	0,7938	0,2491	0,5002	0,0446
25	0,4643	0,4043	0,2757	0,4323	0,6382	0,4239	0,1718	0,0425
26	0,3426	0,3715	0,2169	0,3565	0,8470	0,4944	0,1342	0,2185
27	0,3288	0,3099	0,1898	0,3191	0,8150	0,5079	0,1822	0,1249
28	0,4503	0,2719	0,2042	0,3391	0,7468	0,3132	0,4149	0,0187
29	0,2002	0,3254	0,1415	0,2479	1,3357	0,6387	0,0359	0,6611
30	0,4185	0,4829	0,2890	0,4484	0,7595	0,4286	0,0885	0,2424
31	0,3677	0,2606	0,1799	0,3050	0,7759	0,4117	0,3277	0,0365
32	0,5442	0,5311	0,3676	0,5376	0,5879	0,3258	0,1431	0,1190
33	0,5028	0,3705	0,2712	0,4267	0,6724	0,3234	0,3061	0,0430
34	0,4357	0,4596	0,2881	0,4473	0,6928	0,4429	0,0974	0,1525
35	0,4734	0,4612	0,3048	0,4672	0,6659	0,3860	0,1528	0,1270
36	0,4775	0,4714	0,3110	0,4745	0,6485	0,3959	0,1327	0,1199
37	0,5372	0,4527	0,3257	0,4914	0,5754	0,3618	0,1855	0,0282
38	0,3827	0,4702	0,2674	0,4220	0,8702	0,4181	0,1117	0,3404
\bar{x}	0,4711	0,4017	0,2777	0,4293	0,6980	0,3705	0,2278	0,0997

Tabela 6.6: Uspešnost Klapurijevega algoritma, kadar je utežitvena funkcija w_2 linearna.

Poglavje 7

Zaključek

V tej diplomski nalogi smo se ukvarjali s problemom avtomatične transkripcije večglasnega ljudskega petja. Najprej smo se v grobem spoznali z zvokom in človeškim slušnim sistemom. Podali smo nekaj teorije iz področja digitalnega procesiranja signalov s poudarkom na kratkočasovni frekvenčni analizi. Spoznali smo splošen algoritem za iskanje sočasnih tonov v zvočnem posnetku, ki deluje v frekvenčni domeni in deloma posnema človeški slušni sistem. Pripravili smo testno množico slovenskih ljudskih pesmi, ki so bile posnete na terenu pri pristnih ljudskih pevcih. Transkripcije pesmi smo opravili ročno. Za boljše razumevanje delovanja algoritma za detekcijo tonov in lažjo analizo napak smo razvili interaktivno grafično aplikacijo. Nad množico ljudskih pesmi smo preizkusili algoritem in ga poskusili izboljšati.

Glavni pridobitvi tega dela sta testna množica slovenskih ljudskih pesmi in interaktivna vizualizacija. Poskusi izboljšav algoritma niso prinesli novih odkritij. Zaradi dobrih rezultatov nekoliko mehkejšega obravnavanja napak sta možna dva scenarija: bodisi je potrebno v algoritmu boljše frekvenčno lokalizirati prepoznane tone bodisi je potrebno izboljšati ročne transkripcije testne množice. Ponuja se tudi ideja za novo mero ocenjevanja posnetkov s problematično intonacijo, še zlasti, kadar se na osnovi ocene postavlja kriterij pri metodah strojnega učenja.

Pri izboljšavah algoritma je na voljo še veliko možnosti. Čeprav ni ja-

sno, kaj bi prispevalo k boljši detekciji, pa so se pokazale nekatere smernice. Eno smo že omenili in je boljša frekvenčna lokalizacija tonov, ki jo lahko dosežemo na več načinov, npr. z opazovanjem frekvenc harmoničnih delnih tonov ali pa s pomočjo faznega spektra. Druge možnosti so nadaljnje študije učinka spektralnega beljenja in utežitvenih funkcij. Pri tem bi opazovali kakovost detekcije nad enostavnimi primeri in postopoma na sestavljenih, kompleksnejših. Tretja možnost je vključitev muzikoloških modelov, saj že pri trenutnih rezultatih tonaliteto uspešno prepoznamo na relativno kratkem segmentu prepoznanih tonov in s preprostimi statističnimi metodami. Vendar pa se pri muzikoloških modelih dela še večje predpostavke kot prej na nižjih nivojih, zato tovrstne rešitve niso vedno zaželeno.

Literatura

- [1] R. N. Bracewell, *The Fourier Transform and Its Applications*, McGraw-Hill Higher Education, 2000.
- [2] S. Butterworth, “On the Theory of Filter Amplifiers”, *Wireless Engineer*, zv. 7, 1930.
- [3] S. Dixon, G. Widmer, “MATCH: A Music Alignment Tool Chest”, *6th International Conference on Music Information Retrieval, ISMIR*, sept. 2005.
- [4] K. Dressler, “Pitch Estimation by the Pair-Wise Evaluation of Spectral Peaks”, v zborniku *AES 42th International Conference*, Ilmenau, Germany, julij 2011.
- [5] T. Eerola, P. Toivainen, *MIDI Toolbox: MATLAB Tools for Music Research*, Univerza v Jyväskylä na Finskem, 2004. Dostopno na <http://www.jyu.fi/musica/miditoolbox>. [Dostopano 3. 2. 2015].
- [6] B. R. Glasberg, B. C. J. Moore, “Derivation of auditory filter shapes from notched-noise data”, *Hearing research*, št. 1–2, zv. 47, str. 103–138, 1990.
- [7] M. Golež Kaučič, M. Klobčar, Z. Kumer, U. Šivic, M. Terseglav, *Slovenske ljudske pesmi*, ZRC SAZU, št. 5, 2007.
- [8] A. Klapuri, “Multiple fundamental frequency estimation by summing harmonic amplitudes”, *ISMIR*, 2006, str. 216–221.

-
- [9] A. Klapuri, M. Davy, *Signal Processing Methods for Music Transcription*, Springer, 2006.
- [10] R. G. Lyons, *Understanding Digital Signal Processing*, Pearson Education, 2004.
- [11] U. Michels (prevod P. Kuret, E. Škulj), *Glasbeni atlas*, DZS, 2002, 1. izdaja.
- [12] B. C. J. Moore, B. Glasberg, “Suggested formulae for calculating auditory-filter bandwidths and excitation patterns”, *The Journal of the Acoustical Society of America*, št. 3, zv. 74, str. 750–753, 1983.
- [13] A. V. Oppenheim, R. W. Schaffer, J. R. Buck, *Discrete-time signal processing*, Prentice Hall, 1999.
- [14] T. W. Parks, C. S. Burrus, *Digital filter design*, Wiley, 1987.
- [15] G. E. Poliner, D. P. W. Ellis, “A discriminative model for polyphonic piano transcription”, *EURASIP Journal of Advances in Signal Processing*, št. 1, str. 154, 2007.
- [16] J. G. Roederer, *The Physics and Psychophysics of Music*, Springer, 2008, 4. izdaja.
- [17] J. O. Smith III, J. S. Abel, “Bark and ERB Bilinear Transforms”, *Center for Computer Research in Music and Acoustics (CCRMA)*, Stanford University, 1999. Dostopno na: <https://ccrma.stanford.edu/~jos/bbt/>. [Dostopano 11. 2. 2015].
- [18] T. Tolonen, M. Karjalainen, “A computationally efficient multipitch analysis model”, *IEEE Transactions on Speech and Audio Processing*, št. 6, zv. 8, str. 708–716, 2000.
- [19] R. Turetsky, D. Ellis, “Ground-Truth Transcriptions of Real Music from Force-Aligned MIDI Syntheses”, *4th International Symposium on Music Information Retrieval, ISMIR*, okt. 2003.

-
- [20] E. Zwicker, G. Flottorp, S. S. Stevens, “Critical Band Width in Loudness Summation”, *The Journal of the Acoustical Society of America*, št. 5, zv. 29, 1957.
- [21] Butterworth filter. Dostopno na:
http://en.wikipedia.org/wiki/Butterworth_filter. [Dostopano 10. 2. 2015].
- [22] Cubase — the DAW that puts creativity first. Dostopno na:
http://www.steinberg.net/en/products/cubase/why_cubase.html. [Dostopano 27. 2. 2015].

Dodatek A

Butterworthov filter

V elektro inženiringu so že desetletja pred prihodom računalnikov uporabljali filtre. Teorija o načrtovanju filtrov se je oblikovala kasneje. S prihodom računalnikov in njihovega razvoja do te mere, da so postali bolj uporabni, so inženirji mnoge ideje poskušali prenesti v njihov diskretni svet. Tako je potreba po načrtovanju digitalnih filtrov nastala po tem, ko so bile metode načrtovanja zvezno-časovnih filtrov že utečene. Še danes sta impulzna invarianca in bilinearna transformacija primera metod za načrtovanje digitalnih NEO filtrov, ki se zanašata na načrtovanje filtra v zvezno-časovni domeni. Ta razdelek opisuje en primer aproksimacije idealnega nizkoprepustnega filtra.

Leta 1930 je britanski inženir in fizik Stephen Butterworth objavil članek [2], v katerem je zapisal trditev, da idealni električni filter naj ne bi samo v celoti zavrgel neželene frekvence, temveč bi tudi moral imeti enakomerno občutljivost za želene frekvence. Takšen idealni filter seveda ne obstaja, se je pa temu zadovoljivo približal z opisanim filtrom. Bil je znan po tem, da se je loteval nemogočih matematičnih problemov in je za tisti čas ob pomanjkanju teorije o načrtovanju filtrov, ki jo poznamo danes, potreboval precej inženirskih izkušenj. Zato ni nič čudnega, da je od objave do splošne uporabe njegovega filtra trajalo več kot 30 let [21].

A.1 Magnitudni odziv Butterworthovega filtra

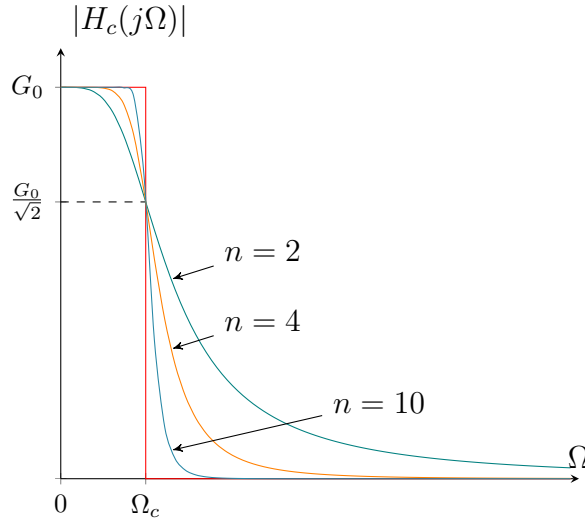
Butterworthov filter sodi v skupino NEO filtrov in je najenostavnejši izmed štirih klasičnih filtrov te skupine. Preostali trije so še Čebišev filter tipov ena in dva ter eliptični. Magnitudni odziv Butterworthovega filtra v zvezni domeni je podan s funkcijo

$$|H_c(j\Omega)| = \sqrt{\frac{G_0^2}{1 + \left(\frac{\Omega}{j\Omega_c}\right)^{2n}}}. \quad (\text{A.1})$$

G_0 je začetna amplituda (pri $\Omega = 0$). $\Omega_c > 0$ označuje frekvenco rezanja, magnituda pri tej frekvenci pa je enaka $\frac{G_0}{\sqrt{2}}$. Funkcija $|H_c(j\Omega)|$ je za $0 \leq \Omega < \infty$ monotonno padajoča in ima v frekvenčno prepustnem delu najbolj raven magnitudni odziv, saj je prvih $2n - 1$ odvodov kvadrata magnitudnega odziva ($|H_c(j\Omega)|^2$) pri $\Omega = 0$ enakih nič. Red filtra je $n \in \mathbb{N}$. Z večanjem reda se filter približuje idealnemu, a za ceno več računanja pri načrtovanju in filtriranju. Pri frekvenci rezanja ($\Omega = \Omega_c$) je naklon magnitudnega odziva filtra enak $-\frac{n}{\sqrt{8}\Omega_c}$, kar pomeni, da filter doseže idealni (navpični) naklon, kadar $n \rightarrow \infty$. V točkah $\Omega = 0$ in $\Omega = \infty$ filter najboljše aproksimira idealni filter, najslabše pa okrog točke $\Omega = \Omega_c$. Magnitudni odziv filtra je gladek na celotnem frekvenčnem območju $\Omega \geq 0$, gladka je tudi faza, a ni linearna. Slika A.1 prikazuje magnitudni odziv v odvisnosti od n .

A.2 Prevajalna funkcija, ničle in poli

Izrazimo prevajalno funkcijo H_c z Laplacovo kompleksno spremenljivko s . Do sedaj je bila H_c izražena s točkami na imaginarni osi $s = j\Omega$, kar je frekvenčni odziv filtra, ali drugače, Fourierjeva transformacija impulznega odziva filtra pri realnih frekvencah Ω . Prevajalna funkcija je racionalna funkcija (kvocient polinomov kompleksne spremenljivke s). Za pasovno omejene filtre mora



Slika A.1: Magnitudni odziv Butterworthovega filtra v odvisnosti od reda n . Ne glede na red filtra je vrednost pri frekvenci Ω_c enaka $\frac{G_0}{\sqrt{2}}$. Polna rdeča črta označuje idealni nizkoprepustni filter.

biti polinom v imenovalcu višje stopnje kot v števcu. Za lažje nadaljnje računanje [14, str. 161] uvedemo novo funkcijo kompleksne spremenljivke s , $\mathcal{H}_c(s) = H_c(s)H_c(-s)$, ki je za točke na imaginarni osi kvadrat magnitudnega odziva filtra: $\mathcal{H}_c(j\Omega) = |H_c(j\Omega)|^2$. Torej je

$$\mathcal{H}_c(s) = \frac{G_0^2}{1 + \left(\frac{s}{j\Omega_c}\right)^{2n}} = \frac{G_0^2}{1 + \left(\frac{-s^2}{\Omega_c^2}\right)^n}. \quad (\text{A.2})$$

Funkcija (A.2) ima $2n$ polov (tj. ničel polinoma v imenovalcu) in eno $2n$ -kratno ničlo (tj. ničel polinoma v števcu) pri $s \rightarrow j\infty$. Pole izračunamo kot rešitve enačbe $\left(\frac{s}{j\Omega_c}\right)^{2n} = -1$, ki so

$$\begin{aligned} \left(\frac{s_k}{j\Omega_c}\right)^{2n} &= -1 = e^{j\pi} \\ s_k &= j\Omega_c e^{j\frac{\pi}{2n}(1+2k)} \quad \text{pri } k = 0, 1, \dots, 2n-1 \\ s_k &= \Omega_c e^{j\frac{\pi}{2n}(1+2k+n)} \end{aligned} \quad (\text{A.3})$$

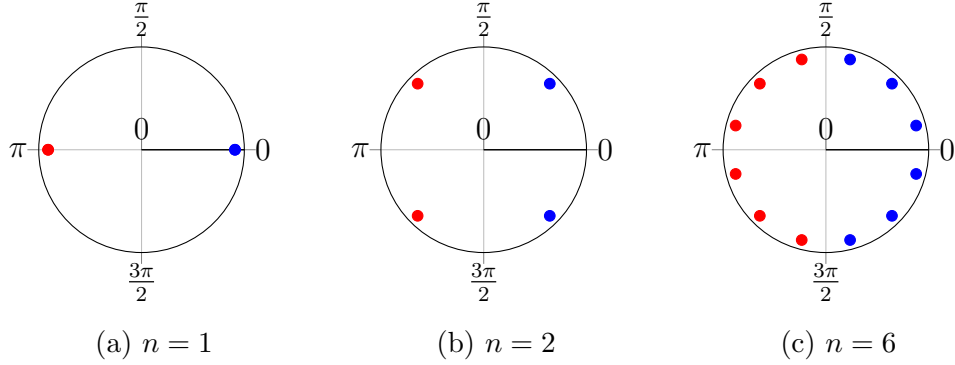
Dobljene rešitve so kompleksna števila, enakomerno razporejena po krožnici s središčem v koordinatnem izhodišču in polmerom Ω_c (slika A.2). Kot med poljubnima sosednima poloma v polarnih koordinatah je konstanten, in sicer $\arg(s_{k+1}) - \arg(s_k) = \frac{\pi}{n}$. Kompleksni poli nastopajo v konjugiranih parih (so simetrični glede na realno os) in velja $s_k = s_{2n-1-k}$ (poli so simetrični glede na imaginarno os). Za doseg stabilnega filtra racionalno prevajalno funkcijo $H_c(s)$ v imenovalcu zgradimo iz vseh n polov, ki imajo negativni realni polravnini, v števcu pa iz n ničel. Preostalih n ničel in n polov (ti imajo pozitivno realno komponento) sestavlja $H_c(-s)$. Če bi prevajalno funkcijo sestavili iz drugih n polov, bi filter lahko imel neskončen impulzni odziv. Oba polinoma prevajalne funkcije predstavimo v ničelni obliki. Magnitudni odziv Butterworthovega nizkoprepustnega filtra je monotona funkcija, ki se približuje ničli, ko gre $s \rightarrow j\infty$, zato ničlo doseže šele v neskončnosti. Posledično ima prevajalna funkcija v števcu samo konstanto, v imenovalcu pa polinom stopnje n :

$$H_c(s) = G_0 \prod_{k=0}^{n-1} \frac{\Omega_c}{s - s_k} = G_0 \prod_{k=0}^{n-1} \frac{\Omega_c}{s - e^{j\frac{\pi}{2n}(1+2k+n)}}. \quad (\text{A.4})$$

Ni težko preveriti, da imajo poli za $k \in \{0, 1, \dots, n-1\}$ negativno realno komponento (fazni koti polov ležijo med $\frac{\pi}{2}$ in $\frac{3\pi}{2}$), torej dobljena prevajalna funkcija $H_c(s)$ zagotavlja stabilen filter.

A.3 Razvoj Butterworthovega magnitudnega odziva

V tem razdelku pokažemo, kako iz magnitudnega odziva idealnega filtra v procesu aproksimacije pridelamo kvadrat magnitudnega odziva Butterworthovega filtra, tj. (A.2). Zaradi enostavnosti predpostavimo, da želimo aproksimirati magnitudni odziv nizkoprepustnega idealnega filtra pri $G_0 = 1$ in $\Omega_c = 1$. Magnitudni odziv idealnega filtra je z rdečo črto prikazan na sliki



Slika A.2: Poli funkcije (A.2) ležijo v s ravnini enakomerno razporejeni po krožnici s polmerom Ω_c . Prikazani so poli pri $n \in \{1, 2, 6\}$. Za stabilen filter v prevajalno funkcijo vključimo samo pole, ki ležijo v polravnini levo od imaginarne osi ($\Re(s) < 0$). Na sliki so označeni z rdečo barvo.

A.1). Ker je končni rezultat aproksimacije enak (A.2), aproksimacijsko funkcijo označimo kar z $\mathcal{H}_c(s)$. Vemo, da je $\mathcal{H}_c(s)$ racionalna funkcija in da je splošna oblika kvadrata magnitudnega odziva soda funkcija realnega argumenta Ω [14], zato lahko \mathcal{H}_c izrazimo kot

$$\mathcal{H}_c(j\Omega) = \frac{\sum_{\ell=0}^m a_{2\ell} \Omega^{2\ell}}{\sum_{\ell=0}^n b_{2\ell} \Omega^{2\ell}}. \quad (\text{A.5})$$

Koeficiente a_0, \dots, a_{2m} in b_0, \dots, b_{2n} lahko izračunamo iz razvoja kvadrata magnitudnega odziva idealnega filtra (označimo ga s \mathcal{H}_i) v Taylorjevo vrsto okrog točke $\Omega = 0$, ki je

$$\mathcal{H}_i(j\Omega) = \sum_{\ell=0}^{\infty} \frac{\mathcal{H}_i^{(\ell)}(0)}{\ell!} \Omega^\ell \text{ in}$$

razvoja H_c v Taylorjevo vrsto okrog $\Omega = 0$ ter primerjavo členov obeh vrst. $\mathcal{H}_i^{(\ell)}(0)$ označuje ℓ -ti odvod funkcije \mathcal{H}_i v točki nič. Funkcija \mathcal{H}_i je v točki nič neskončnokrat odvedljiva, a vsi odvodi razen pri $\ell = 0$ so nič (odvod konstantne funkcije). Da dobimo nizkoprepustni filter, določimo $\mathcal{H}_c(j0) = 1$, od koder sledi $a_0 = b_0$ in $\mathcal{H}_c(j\infty) = 0$, ki se prevede na pogoja $m < n$ in $b_{2n} \neq 0$. Z nekaj algebrائيčne telovadbe izračunamo odvode $\mathcal{H}_c^{(\ell)}(0)$ za vse

$\ell \in \{1, 2, \dots, 2n - 1\}$ in jih enačimo z nič. Dobimo naslednji sistem enačb:

$$\begin{aligned} b_{2\ell} &= a_{2\ell}, & \ell &\in \{0, 1, \dots, m\}, \\ b_{2\ell} &= 0, & \ell &\in \{m + 1, \dots, n - 1\}, \\ b_{2n} &\neq 0. \end{aligned}$$

Rešitev sistema je $(m + 2)$ -parametrična in zato ni enolična. Sledi, da lahko koeficiente števca za optimalno aproksimacijo (A.5) izberemo poljubno, prvih $m + 1$ koeficientov imenovalca nastavimo enako kot v števcu in izberemo še neničelni koeficient pri najvišji potenci v imenovalcu. Če je $a_0 = 1$, $a_{2\ell} = 0$ za $\ell > 0$ in $b_{2n} = 1$, se dobljena $\mathcal{H}_c(j\Omega)$ ujema z (A.2) pri $G_0 = \Omega_c = 1$. V splošnem koeficient b_{2n} predstavlja obrat frekvence rezanja ($b_{2n} = \Omega_c^{-1}$).

Butterworthov filter je aproksimacija idealnega filtra s pomočjo Taylorjeve vrste pri $\Omega = 0$ in $\Omega \rightarrow \infty$. Izmed klasičnih NEO filtrov Čebiševa filtra tipov ena in dva ter eliptični NEO filter poleg aproksimacije s Taylorjevo vrsto uporabljajo še Čebiševo aproksimacijo [14].

A.4 Primer načrtovanja digitalnega filtra

Pri aplikacijah, ki v magnitudnem odzivu v nobenem frekvenčnem območju ne dopuščajo valovanja, lahko uporabimo Butterworthov filter. Za zvezno-časovne signale lahko pole in sistemsko funkcijo analognega filtra izračunamo po postopku iz razdelka A.2, sistemsko funkcijo digitalnega Butterworthovega filtra pa lahko pridobimo s transformacijo sistemske funkcije analognega filtra v z ravnino. Preslikavo s ravnine v z ravnino lahko opravimo z metodo impulzne invariance ali pa z bilinearne transformacije. V tem razdelku je podan primer načrtovanja nizkoprepustnega digitalnega Butterworthovega filtra s pomočjo bilinearne transformacije. Izhodišče so podane specifikacije iz razdelka 3.2, kjer je potrebno skonstruirati filter za potrebe podvzorčenja.

Naj ω označuje frekvenco v radianih na vzorec. Specifikacije 3.2, pretvorjene v radialno frekvenco po formuli $f_s \omega = f 2\pi$ pri $f_s = 44,1$ vzorcev na

sekundo in frekvenci f v Hz, so naslednje:

$$1 \geq |H(e^{j\omega})| \geq 0,9477, \quad 0 \leq \omega \leq \frac{2\pi}{7}, \quad (\text{A.6a})$$

$$0 \leq |H(e^{j\omega})| \leq 0,0371, \quad \frac{\pi}{2} \leq \omega \leq \pi. \quad (\text{A.6b})$$

Sistemska funkcijo $H(z)$ digitalnega filtra izračunamo tako [13]:

1. Kritične frekvence diskretno-časovnega filtra iz specifikacij preslikamo v ustrezne frekvence zvezno-časovnega filtra.
2. Iz novih specifikacij in znane analitične formule kvadrata magnitudnega odziva Butterworthovega filtra izračunamo najmanjši možni red filtra, ki mora biti celo število, in frekvenco rezanja.
3. Izračunamo pole in sistemsko funkcijo $H_c(s)$.
4. $H_c(s)$ z bilinearno transformacijo preslikamo v $H(z)$.

Kritične frekvence diskretnočasovnega filtra iz specifikacij (A.6) pretvorimo v frekvence zveznočasovnega filtra po formuli $\Omega = \tan(\frac{\omega}{2})$, kot to zahteva metoda bilinearne transformacije. Nove specifikacije so

$$1 \geq |H_c(j\Omega)| \geq 0,9477, \quad 0 \leq \Omega \leq 0,481575, \quad (\text{A.7a})$$

$$0 \leq |H_c(j\Omega)| \leq 0,0371, \quad 1 \leq \Omega \leq \infty. \quad (\text{A.7b})$$

V primerjavi s specifikacijami (A.6) je spremenjeno le frekvenčno območje, in sicer iz intervala $[0, \pi]$ v interval $[0, \infty)$. Zaradi monotonosti magnitudnega odziva Butterworthovega filtra se specifikacije poenostavijo v

$$|H_c(j 0,481575)| \geq 0,9477, \quad (\text{A.8a})$$

$$|H_c(j 1)| \leq 0,0371. \quad (\text{A.8b})$$

Na podlagi neenakosti (A.8) določimo najmanjši red filtra in frekvenco rezanja. V specifikacijah (A.8) neenakosti nadomestimo z enakostma, dobljeni

enačbi kvadriramo, za $|H_c(\cdot)|^2$ pa uporabimo (A.1) pri $G_0 = 1$. Ostane nam

$$\frac{1}{1 + \left(\frac{0,481575}{\Omega_c}\right)^{2n}} = 0,9477^2, \quad (\text{A.9a})$$

$$\frac{1}{1 + \left(\frac{1}{\Omega_c}\right)^{2n}} = 0,0371^2. \quad (\text{A.9b})$$

Rešimo sistem. Najprej eliminiramo Ω_c in izračunamo red:

$$n = \left\lceil \frac{\log(0,9477^{-2} - 1) - \log(0,0371^{-2} - 1)}{2 \log 0,481575} \right\rceil \approx [5,99675] = 6.$$

Red filtra n vstavimo v eno izmed enačb (A.9) in izračunamo Ω_c . Po enačbi (A.9a) dobimo

$$\Omega_c = \left(\frac{0,481575}{0,9477^{-2} - 1} \right)^{\frac{1}{12}} \approx 0,577352, \quad (\text{A.10})$$

po enačbi (A.9b) pa $\Omega_c \approx 0,57758$. Izračunani vrednosti frekvence rezanja se razlikujeta zaradi zahteve, da je red n naravno število. Z izbiro enačbe (A.9a) so specifikacije v območju prepustnosti natančno izpolnjene, v območju neprepustnosti pa so nekoliko prekoračene. Za Ω_c , izračunanem po (A.9b), je situacija ravno obratna. V nadaljevanju za frekvenco rezanja uporabimo vrednost (A.10).

Ko sta red filtra in frekvenca rezanja izračunana, poznamo funkcijo $\mathcal{H}_c(s)$ v s ravnini (A.2). Od tod poiščemo sistemsko funkcijo $H_c(s)$ analognega filtra. V ta namen izračunamo vseh $2n$ ničel in $2n$ polov funkcije $\mathcal{H}_c(s)$. Vse ničle se nahajajo pri $s \rightarrow \infty$, poli pa so enakomerno razporejeni po krožnici s središčem v izhodišču in polmerom (A.10) in jih izračunamo po (A.3). Vseh 12 polov funkcije $\mathcal{H}_c(s)$ je prikazanih na sliki A.2c. Ker so ničle in poli izračunani iz kvadrata magnitudnega odziva sistemske funkcije $H_c(s)$, vanjo vključimo samo polovico ničel in polovico polov. Za stabilen filter izberemo pole, ki imajo negativno realno komponento (na sliki A.2c so obarvani rdeče).

Če konjugirane pare polov združimo skupaj, je $H_c(s)$ po (A.4) pri $G_0 = 1$ enaka

$$\begin{aligned} H_c(s) &= \prod_{k=0}^5 \frac{\Omega_c}{s - \Omega_c e^{j \frac{\pi}{12}(7+2k)}} \\ &= \frac{0,0370377}{(s^2 + 0,298859s + 0,333335)(s^2 + 0,816499s + 0,333335)} \\ &\quad \cdot \frac{1}{(s^2 + 1,11536s + 0,333335)}. \end{aligned} \quad (\text{A.11})$$

Ostane še preslikava sistemske funkcije $H_c(s)$ iz s ravnine v z ravnino. Definicija preslikave je

$$s = \frac{1 - z^{-1}}{1 + z^{-1}}, \quad (\text{A.12})$$

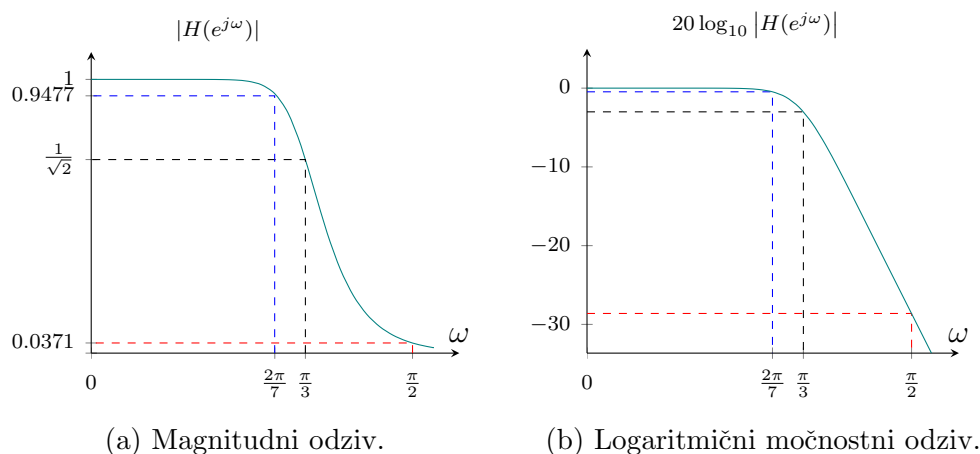
ki navpične premice v kompleksni s ravnini (vzporednice z imaginarno osjo) preslika v krožnice z izhodiščem v koordinatnem izhodišču v kompleksni z ravnini. Končno, (A.12) vstavimo v (A.11), preoblikujemo v

$$\begin{aligned} H(z) &= \frac{S(z)}{I(z)}, \quad S(z) = 0,004311 + 0,025863z^{-1} + 0,064658z^{-2} \\ &\quad + 0,086211z^{-3} + 0,064658z^{-4} + 0,025863z^{-5} + 0,004311z^{-6} \\ I(z) &= 1 - 1,9816z^{-1} + 2,2524z^{-2} - 1,4693z^{-3} + 0,5963z^{-4} \\ &\quad - 0,1354z^{-5} + 0,0136z^{-6} \end{aligned} \quad (\text{A.13})$$

in s tem dobimo sistemsko funkcijo digitalnega filtra $H(z)$ za podane specifikacije (A.6). Frekvenca rezanja (A.10) je v diskretni domeni enaka

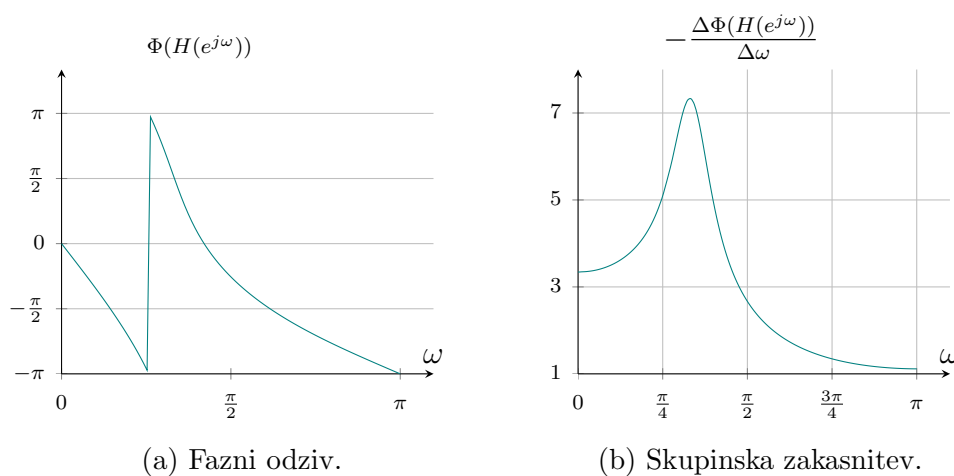
$$\omega_c = 2 \arctan(\Omega_c) = 2 \arctan(0,577352) \approx 1,0472 \approx \frac{\pi}{3}. \quad (\text{A.14})$$

Ali pravkar izračunan filter zadošča specifikacijam? Sliki A.3a in A.3b prikazujeta magnitudni in logaritmični močnostni odziv filtra. Na grafih je označena meja med prepustnim in prehodnim delom ter prehodnim in neprepustnim delom, kot to zahtevajo specifikacije. Razvidno je, da filter tem



Slika A.3: Magnitudni in logaritmični močnostni odziv Butterworthovega digitalnega filtra $H(z)$ za specifikacije (A.6). Na slikah A.3a in A.3b modra črtkana navpična črta označuje mejo prepustnosti, vodoravna pa faktor največjega zmanjšanja magnitude v prepustnem delu. Podobno rdeča navpična črta označuje mejo neprepustnosti po specifikacijah, vodoravna pa najmanjši faktor zadušitve neželenih frekvenc. Črna črtkana črta označuje frekvenco rezanja in magnitudno vrednost pri tej frekvenci.

omejitvam zadostuje. Kako pa je s prekrivanjem? V razdelku A.2 je rečeno, da magnitudni odziv zvezno-časovnega Butterworthovega filtra doseže ničlo pri $s \rightarrow j\infty$. Bilinearna transformacija frekvence iz zvezne domene na intervalu $[0, \infty)$ v diskretni domeni skrči na interval $[0, \pi)$, zato je v diskretni domeni ničla pri $z = e^{j\pi} = -1$. Magnitudni odziv se v diskretni domeni periodično ponavlja s periodo 2π . Posamezen cikel neničelnega odziva se razteza na intervalu $((2k-1)\pi, (2k+1)\pi)$, pri $k \in \mathbb{Z}$, zato med ponovitvami odziva ni prekrivanja. Žal pa to še ne pomeni, da lahko pri načrtovanju digitalnih filtrov z bilinearno transformacijo na ta problem pozabimo. Spomnimo, da je filter (A.13) bil uporabljen za potrebe podvzorčenja signala pri faktorju podvzorčenja dva. Po filtriranju signala x se frekvenca podvzorčenja razpolovi. Kaj se zgodi, lahko vidimo na sliki A.5. V okolici frekvence $\frac{\pi}{2}$ pride do prekrivanja, vendar se njegov vpliv v pasu prepustnosti skoraj povsem izniči, zato lahko sklepamo, da je bil filter primerno načrtovan.



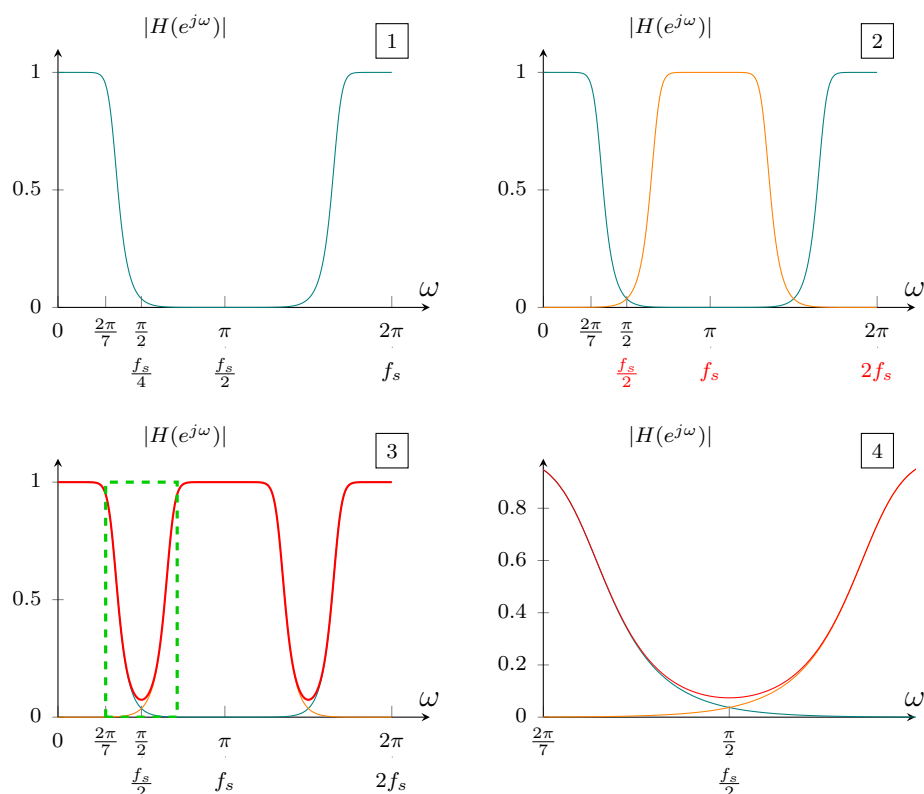
Slika A.4: Fazni odziv in skupinska zakasnitev Butterworthovega digitalnega filtra $H(z)$ za specifikacije (A.6).

A.5 Faza Butterworthovega filtra

Skozi celotno poglavje o Butterworthovem filtru A je bilo do sedaj le malo besed namenjeno fazi, saj načrtovanje filtra, kot opisano zgoraj, upošteva samo magnitudni odziv filtra. Fazni odziv filtra vpliva na fazo izhodnega filtriranega signala [10]. Kadar je faza filtra neničelna, je izhodni signal iz filtra v primerjavi z vhodnim časovno zamaknjen. Idealni filter bi imel linearno fazo in konstantno skupinsko zakasnitev, kar pomeni, da imajo vse frekvence na izhodu iz idealnega digitalnega filtra enako časovno zakasnitev, medtem ko imajo različne frekvence na izhodu iz filtra z nelinearno fazo različno časovno zakasnitev. Takšen idealen filter obstaja — vsak KEO filter z (anti)simetričnimi koeficienti (tj. impulznim odzivom) je idealen v tem pogledu [13].

Povejmo še, kakšen je fazni odziv dobljenega Butterworthovega filtra (A.11). Iz grafov A.4a in A.4b je razvidna nelinearnost faze. Prvi graf prikazuje fazo v radianih, drugi pa skupinsko zakasnitev (enota je število vzorcev), ki je odvod faze po frekvenci. V primeru linearne faze je skupinski zamik konstanten (v grafu bi bila ravna vodoravna črta). Iz grafa A.4b lahko razberemo, da je izhodni signal iz filtra (A.11) v primerjavi z vhodnim pri

frekvenci $\frac{\pi}{4}$ zakasnjjen približno pet vzorcev, pri frekvenci $\frac{\pi}{2}$ je zakasnjjen tri vzorce in pri frekvenci π en vzorec. Tej težavi se lahko izognemo, če filtriramo v obe smeri [10], saj tako izničimo vpliv faze filtra.



Slika A.5: Grafi prikazujejo problem prekrivanja, če po filtriranju frekvenco vzorčenja razpolovimo. Graf 1 prikazuje magnitudni odziv digitalnega filtra (A.13), ki je periodičen s periodo 2π . Na grafu 2 se zaradi razpolovitve frekvence vzorčenja razpolovi tudi perioda. V okolici frekvence $\frac{\pi}{2}$ se sosednja magnitudna odziva prekrijeta. Na grafu 3 je s krepko rdečo črto prikazan dejanski magnitudni odziv po razpolovitvi frekvence vzorčenja (prekriti deli se seštejejo). Vsebina grafa 3 znotraj zelenega okvirčka je približana na grafu 4, kjer je razvidno, da se vpliv prekrivanja v prehodnem delu do pasu prepustnosti pri $\frac{2\pi}{7}$ skoraj povsem izniči.